LNCS 3976

Fernando Boavida Thomas Plagemann
Burkhard Stiller Cedric Westphal
Edmundo Monteiro (Eds.)

NETWORKING 2006

Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems

5th International IFIP-TC6 Networking Conference
Coimbra, Portugal, May 2006
Proceedings

ifip

Springer

Lecture Notes in Computer Science 3976

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Fernando Boavida Thomas Plagemann
Burkhard Stiller Cedric Westphal
Edmundo Monteiro (Eds.)

NETWORKING 2006
Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems

5th International IFIP-TC6 Networking Conference
Coimbra, Portugal, May 15-19, 2006
Proceedings

Springer

Volume Editors

Fernando Boavida
Edmundo Monteiro
Universidade de Coimbra
Departamento de Engenharia Informatica, Polo II
Pinhal de Marrocos, 3030-290 Coimbra, Portugal
E-mail:{boavida,edmundo}@dei.uc.pt

Thomas Plagemann
Universitetet i Oslo
Institutt for informatikk
Postboks, 1080 Blindern, 0316 Oslo, Norway
E-mail: plageman@ifi.uio.no

Burkhard Stiller
University of Zürich
Institut für Informatik
Winterthurerstr. 190, 8057 Zürich, Switzerland
E-mail: stiller@tik.ee.ethz.ch

Cedric Westphal
Nokia
313 Fairchild dr., Mountain View, CA 94043, USA
E-mail: cedric.westphal@noika.com

Library of Congress Control Number: 2006925173

CR Subject Classification (1998): C.2, C.4, H.4, D.2, J.2, J.1, K.6, K.4

LNCS Sublibrary: SL 5 – Computer Communication Networks and Telecommunications

ISSN 0302-9743
ISBN-10 3-540-34192-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-34192-5 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: 11753810 06/3142 5 4 3 2 1 0

Preface

Networking 2006 was organized by the University of Coimbra, Portugal, and it was the fifth event in a series of International Conferences on Networking sponsored by the IFIP Technical Committee on Communication Systems (TC 6). Previous events were held in Paris (France) in 2000, Pisa (Italy) in 2002, Athens (Greece) in 2004, and Waterloo (Canada) in 2005.

Networking 2006 brought together active and proficient members of the networking community, from both academia and industry, thus contributing to scientific, strategic, and practical advances in the broad and fast-evolving field of communications.

The conference comprised highly technical sessions organized thematically, keynote talks, tutorials offered by experts, as well as workshops and panel discussions on topical themes. Plenary sessions with keynote talks opened the daily sessions, which covered Networking Technologies, Services and Protocols, Performance of Computer and Communication Networks, and Mobile and Wireless Communications Systems.

The Networking 2006 call for papers attracted 440 submissions from 44 different countries in Asia, Australia, Europe, North America, and South America. These were subject to thorough review work by the Program Committee members and additional reviewers. The selection process was finalized in a Technical Program Committee meeting held in Lisbon on January 23, 2006.

A high-quality selection of 88 full papers and 31 posters, organized into 24 regular sessions and 1 poster session, made up the Networking 2006 main technical program, which covered wireless networks, mobile ad-hoc networks, sensor networks, optical networks, peer-to-peer topology and location awareness, mobility, traffic engineering, routing, transport protocols, monitoring and measurements, resource management, quality of service, multimedia, and caching and content management. The technical program was complemented by three keynote speeches, by Monique Morrow (Cisco Systems, USA), Costas Courcoubetis (Athens University of Economics and Business, Greece) and Muriel Médard (MIT, USA), on next-generation networking, peer-to-peer systems, and network coding, respectively,

In addition to the main technical program, the day preceding the conference was dedicated to six excellent tutorials on BGP - Interdomain Routing and Virtual Private Networks, IP-Oriented QoS in the Next Generation Networks: Application to Wireless Networks, Extensible IP Signaling: Architecture, Protocols and Practice, Roadmap to Cross-Layer and Cross-System Optimization for B3G, Peer-to-Peer Networking, and User-Directed and QoS-Driven Routing: Theoretical and Experimental Considerations, respectively given by Olivier Bonaventure (Catholic University of Louvain, Belgium), Pascal Lorenz (University of Haute-Alsace, France), Xiaoming Fu and Hannes Tschofenig (University of Goettingen and Siemens, Germany), George Kormentzas and Charalabos Skianis (University of the Aegean Karlovassi and NCSR 'D', Greece), Raouf Boutaba (University of Waterloo, Canada), and Erol Gelenbe (University of Central Florida, Orlando, USA).

The final day of Networking 2006 was dedicated to five one-day workshops on the following topics: Security and Privacy in Mobile and Wireless Networking, Content Caching and Distribution Networks, Performance Control in Wireless Sensor Networks, Towards the QoS Internet, and Next-Generation Networking Middleware.

We wish to record our appreciation of the efforts of many people in bringing about the Networking 2006 conference: to all the authors that submitted their papers to the conference, regretting that it was not possible to accept more papers; to the Program Committee and to all associated reviewers; to our sponsors and supporting institutions. Finally, we would like to thank all the people that helped us at the University of Coimbra, namely, Márcia Espírito Santo, Paula Mano, and all the volunteers from the Laboratory of Communications and Telematics.

May 2006

Fernando Boavida
Thomas Plagemann
Burkhard Stiller
Cedric Westphal
Edmundo Monteiro

Organization

Executive Committee

General Chair: Edmundo Monteiro, University of Coimbra, Portugal

Program Chair: Fernando Boavida, University of Coimbra, Portugal

Chair, Special Track for Networking Technologies, Services and Protocols:
Thomas Plagemann, University of Oslo, Norway

Chair, Special Track for Performance of Computer & Communication Networks:
Burkhard Stiller, University of Zurich and ETH Zurich, Switzerland

Chair, Special Track for Mobile and Wireless Communications:
Cedric Westphal, Nokia, USA

Keynote Chairs:
Guy Leduc, University of Liège, Belgium
Mário Freire, University of Beira Interior, Portugal

Tutorial Chairs:
Nelson Fonseca, Univ. Campinas, Brazil
Paulo Simões, University of Coimbra, Portugal

Workshop Chairs:
Charalabos Skianis, NCSR 'D', Greece
Jorge Sá Silva, University of Coimbra, Portugal

Publicity Chairs:
Bu-Sung Lee, Nanyang Technological University, Singapore
João Orvalho, Polytechnic Institute of Coimbra, Portugal

Industry Relations Chairs:
Biswajit Nandy, Solana Net., Canada
Graça Carvalho, Cisco Europe

Publication Chairs:
Xavier Masip-Bruin, UPC, Spain
Marilia Curado, University of Coimbra, Portugal

Last Edition Chairs:
Jay Black, University of Waterloo, Canada
Raouf Boutaba, University of Waterloo, Canada

Local Organizing Committee

Jorge Sá Silva, University of Coimbra, Portugal
Paulo Simões, University of Coimbra, Portugal
Marilia Curado, University of Coimbra, Portugal
João Orvalho, Instituto Politécnico de Coimbra, Portugal

Steering Committee

Otto Spaniol (Chair), RWTH-Aachen University, Germany
Augusto Casaca, IST/INESC, Portugal
Guy Omidyar, TC6 WG6.8 Chair, USA
Guy Pujolle, University of Paris 6, France
Harry Perros, NCSU, USA
Ioannis Stavrakakis, Univ. Athens, Greece

Supporting and Sponsoring Organizations (alphabetically)

Alcatel
Autoridade Nacional das Comunicações
Associação para a Promoção e Desenvolvimento da Sociedade da Informação
Câmara Municipal de Coimbra
Cisco Systems
Euro NGI Network of Excellence
Fundação Calouste Gulbenkian
Fundação Luso-Americana
Fundação Oriente
Fundação para a Ciência e a Tecnologia, FCT
IFIP TC 6
University of Coimbra

Program Committee

Aaron Striegel, University of Notre Dame, USA
Alexandre Santos, University of Minho, Portugal
Ana Pont, Politechnical University of Valencia, Spain
Andrea Bianco, Politecnico di Torino, Italy
Ashwin Sridharan, Sprint ATL, USA
Athina Markopoulou, University California Irvine, USA
Baochun Li, University of Toronto, Canada
Bhaskar Krishnamachari, USC, USA
Biswajit Nandy, Solana Networks, Canada
Boon Sain Yeo, Institute For Infocomm Research, Singapore
Burkhard Stiller, University of Zurich and ETH Zurich, Switzerland

Bu-Sung Lee, NTU, Singapore
Cedric Westphal, Nokia, USA
Charalabos Skianis, NCSR 'D', Greece
Chris Blondia, University of Antwerp, Belgium
Christos Papadopoulos, University of Southern California, USA
Claudio Casetti, Polytechnic of Torino, Italy
Costas Courcoubetis, Athens University of Econ. and Business, Greece
Daniel Zappala, Brigham Young University, USA
David Hutchison, Lancaster University, UK
Eckhart Koerner, University of Applied Sciences Mannheim, Germany
Edmundo Monteiro, University of Coimbra, Portugal
Eitan Altman, INRIA, France
Erwin Rathgeb, University of Essen, Germany
Eylem Ekici, Ohio State University, USA
Fan Bai, General Motors, USA
Fernando Boavida, University of Coimbra, Portugal
Frank Huebner, AT&T, USA
George Kormentzas, University of the Aegean Karlovassi, Greece
George Rouskas, North Carolina State University, USA
Gerald Maguire, KTH, SE
Gianluca Iannaccone, Intel Research Cambridge, UK
Guenter Haring, University of Vienna, Austria
Guoliang Xue, Arizona State University, USA
Guy Leduc, University of Liege, Belgium
Guy Pujolle, University of Paris 6, France
Harry Perros, North Carolina State University, USA
Ilkka Norros, VTT Technical Research Centre of Finland, Finland
Ioanis Nikolaidis, University of Alberta, Canada
Ivan Stojmenovic, University of Ottawa, Canada
Jaudelice De Oliveira, Drexel University, USA
Jerome Galtier, France Telecom R&D and INRIA, France
Jianping Pan, Victoria University, Canada
Joe Finney, Lancaster University, UK
Jordi Domingo-Pascual, Technical University of Catalunya, Spain
Jorg Liebeherr, University of Toronto, Canada
Jorge Sá Silva, University of Coimbra, Portugal
José Ruela, INESC, Portugal
Josep Sole Pareta, Technical University of Catalunya, Spain
Jun-Hong Cui, University of Connecticut, USA
Katia Obraczka, University of California, Santa Cruz, USA
Ketan Mayer-Patel, University of North Carolina, USA
Kevin Almeroth, University of California at Santa Barbara, USA
Kimmo Raatikainen, University of Helsinki, Finland
Kimon Kontovasilis, NCSR Demokritos, Greece
Konstantinos Psounis, University of Southern California, USA
Lars Wolf, Technical University of Braunschweig, Germany
Laura Feeney, Swedish Institute of Computer Science, SE

Laura Galluccio, University of Catania, Italy
Laurent Mathy, Lancaster University, UK
Luciano Bononi, University of Bologna, Italy
Luis Orozco Barbosa, University of Castilla la Mancha, Spain
Manimaran Govindarasu, Iowa State University, USA
Manuel Ricardo, INESC, Portugal
Marco Conti, IIT-CNR, Italy
Marilia Curado, University of Coimbra, Portugal
Mário Freire, University of Beira Interior, Portugal
Mário Serafim Nunes, Technical University of Lisbon, Portugal
Martin Karsten, University of Waterloo, Canada
Martin Mauve, University of Düsseldorf, Germany
Matthias Frank, University of Bonn, Germany
Maurice Gagnaire, ENST, France
Michael Menth, University of Wuerzburg, Germany
Michel Diaz, LAAS-CNRS, France
Milind Buddhikot, Bell Labs, Lucent Technologies, USA
Nelson Fonseca, University of Campinas, BR
Olivier Bonaventure, Catholic University of Louvain, Belgium
Otto Spaniol, Aachen University of Technology, Germany
Pascal Lorenz, University of Haute-Alsace, France
Paul Amer, University of Delaware, USA
Paulo Carvalho, University of Minho, Portugal
Paulo Mendes, Docomo NTT Eurolabs, Germany
Paulo Pinto, New University of Lisbon, Portugal
Paulo Simões, University of Coimbra, Portugal
Peng Ning, North Carolina State University, USA
Peter Key, Microsoft, UK
Peter Reichl, FTW, Austria
Philippe Owezarski, LAAS-CNRS, France
Piet Van Mieghem, Technical University of Delft, The Netherlands
Prosper Chemouil, France Télécom R&D, France
Ramon Puigjaner, University of Balearic Islands, Spain
Reza Rejaie, University of Oregon, USA
Ronald Addie, University of Southern Queensland, AU
Rong Zheng, University of Houston, USA
Rui Aguiar, University of Aveiro, Portugal
Rui Rocha, Technical University of Lisbon, Portugal
Sebastià Galmés, University of Balearic Islands, Spain
Sergi Sanchez López, Technical University of Catalunya, Spain
Stefano Basagni, Northeastern University, USA
Sung Ju Lee, HP Labs, USA
Susana Sargento, University of Aveiro, Portugal
Teresa Vazão, Technical University of Lisbon, Portugal
Thomas Kunz, Carleton University, Canada
Thomas Plagemann, University of Oslo, Norway
Torsten Braun, University of Bern, Switzerland

Vera Goebel, University of Oslo, Norway
Violet Syrotiuk, Arizona State University, USA
Wojciech Burakowski, Warsaw University of Technology, Poland
Wu-Chi Feng, Portland State University, USA
Xavier Masip-Bruin, Technical University of Catalunya, Spain
Yannis Viniotis, North Carolina State University, USA
Yevgeni Koucheryavy, Tampere University of Technology, Finland
Yingfei Dong, University of Hawaii at Manoa, USA
Yongbing Zhang, University of Tsukuba, Japan
Yongguang Zhang, HRL Labs, USA
Youssef Iraqi, Doha University, Oman

Additional Reviewers

Adam Wolisz
Adriano Moreira
Ahmed Serhrouchni
Alberto Cabellos-Aparicio
Alessio Vecchio
Alex Vorbau
Alexander Zimmermann
Alexandre Fonte
Alexandre Proutière
Aline Viana
Almerima Jamakovic
Andre Rodrigues
Andrea Passarella
Andreas Jungmaier
Andrew Eckford
Aníbal Ferreira
Anton Vinokurov,
Antonio Costa
Antonio Panto
Antonio Pietrabissa
Antonio Pinizzotto
Anujan Varma
Aroon Nataraj
Avadora Dumitrescu
Bartomeu Serra
Belkacem Daheb
Benny Van Houdt
Bensong Chen
Bensong Chen
Bernoulli Thomas
Bill Silverajan,
Bing Wang
Binjie Fu
Björn Scheuermann
Brogle Marc
Bruno Dias
Bruno Quoitin
Carla Raffaelli
Carolina Pinart Gilberga
Catalina M. Llado
Chiara Piglione
Chris Gauthier Dick
Christian Lochert Heinrich
Claudia Eckert
Costas Kalogiros
Cristel Pelsser
Daniel Morato
Daniel Popa
Daniele Miorandi
Danzeisen Marc
David Garduno
David Hausheer
Davide Careglio
Davide Careglio
De Cleyn Peter
Dennis Pfisterer
Dimitrios Pezaros
Dinesh Kumar
Dingbang Xu
Dirk Westhoff
Dmitri Moltchanov
Driss Benhaddou
Driss Benhaddou
Dugeon Olivier
Dugeon Olivier
Eduardo Cerqueira
Eguzki Astiz Lezaun
Eleonora Borgia
Emmanuel Lochin
Enrico Schiattarella
Enzo Mingozzi
Eva Marin Tordera
Fernando Kuipers
Franca Delmastro
Francois Pierre
Ganesha Bhaskara Bhaskara
Ge Xiaohu
George Rouskas
Gigi Karmous-Edwards
Giovanni Turi
Giulio Galante
Gourhant Yves
Guillaume Valadon
Guillermo Rodriguez-Navas
Guillermo Rodríguez-Navas
Guyard Frederic
Gwendal Le Grand
Hajime Nakamura

Hamid Sadjadpour
Hannu Reittu
Hasan Hasan
Hebuterne Gerard
Helmut Hlavacs
Hong Zhou Zhou
Huijuan Wang
Ian Marsh
Idris Rai
Ignacio De Miguel
Jimenez
Ilia Baldine
Injong Rhee
Ioannis Lambadaris
Isabella Cerutti
Iyengar Janardhan
Jabed Faruque
Jamal Hadi Salim
Jan Demeer
Jan Gerke
Jarmo Harju
Jaume Comellas
Jay Boice
Jens Milbrandt
Jeongkeun Lee
Jian Tang
Jian Zhang
Jijun Yin
Jing Teng
Jinhui Shen
Joaquim Macedo
John Leis
Jorge Finochietto
Josep Mangues-Bafalluy
Julien Ridoux
Jumpot Phuritatkul
Kaiqi Xiong
Karim Guennoun
Karin Hummel
Karla Ziri-Castro
Khaldoun Al Agha
Khaled Harfoush
Kun Sun
Lambert Joke
Lambros Sarakis
Letor Nico
Levis Pierre
Li Lao
Li Yang
Lin Zhong
Lisong Xu
Loránd Jakab
Luca Muscariello
Luciana Pelusi
Luis M. Correia
Luis Rojas
Manish Jain
Manos Dramitinos
Marcelo Yannuzzi
Marco Fiore
Marco Liebsch
Marco Mellia
Maria Elena Renda
Maria Joao Nicolau
Maria Solange Rito
Lima
Mário Jorge Leitão
Martin Waldburger
Massimiliano Lenardi
Mathieu Bertrand
Mesut Günes
Meuric Julien
Miguel Angel
Labrador
Mihai Sichitiu
Mikel Izal
Milic Dragan
Miroslaw Klinkoski
Moshe Zukerman
Nabil Seddigh
Natarajan Preethi
Nazanin Magharei
Nicolas Larrieu
Nizar Bouabdallah
Norbert Egi
Olivier Dugeon
Olivier Festor
Olivier Klopfenstein
Oscar Gama
Oscar Gonzalez
Ozan Tongoz
Pai Peng
Panayotis Antoniadis
Paolo Giaccone
Pascal Berthou
Pascal Kurtansky
Paul Smith
Pedro Alipio
Pedro Cuenca
Pedro Estrela
Pedro Nuno Sousa
Pedro Vale Estrela
Peeters Gino
Pere Barlet-Ros
Peter Pieda
Raj Gururajan
Gururajan
Raja Sengupta
Rajendra Persaud
Rajendra Persaud
Ralf Wienzek
Reda Haddad
René Serral-Gracià
Ricardo Martínez
Ricardo Pereira
Richard Barton
Rob Kooij
Roberta Fracchia
Rodolfo Oliveira
Rodrigo Rodrigues
Roman Dunaytsev
Romaszko Sylwia
Rougier Jean-Louis
Ruediger Martin
Rui Prior
Ryan Vogt
Salvatore Spadaro
Santpal Dhillon
Satyajayant Misra
Saverio Mascolo
Scheidegger Matthias
Seongkwan Kim
Sergios Soursos
Shao-Cheng Wang
Shyam Kapadia
Silvia Farraposo
Spaey Kathleen
Sridhar Machiraju
Staub Thomas
Stavros Routzounis

Stefan Diepolder
Taekyoung Kwon
Tamer Elbatt
Teixeira Renata
Thanasis Papaioannou
Thierry Rakotoarivelo
Thomas Bernoulli
Thomas Bohnert
Tiago Camilo
Tianhao Qiu
Tim Seipold
Tom Kleiberg
Tracy Camp
Valeria Baiamonte
Van De Velde Erwin
Van Den Wijgaert Nik
Van Houdt Benny
Van Velthoven Jeroen
Venkatesh Rajendran
Vijay Devarapalli
Vikas Paliwal
Vishwas Puttasubbappa
Voorhaen Michael
Wälchli Markus
Wei-Jen Hsu
Weiyi Zhang
Wenhong Tian
Wenye Wang
Weyland Attila
Wilfried Gansterer
Wissam Fawaz
Wolfgang Kiess
Xenia Mountrouidou
Xiaoming Zhou
Xiaoyan Hong
Xin Liu
Yang Yu
Yong Huat Chew
Yufeng Xin
Yuming Jiang
Yuning He
Yunnan Wu
Zhongwei Zhang Zh

Table of Contents

Mobile Ad-Hoc Networks I

Traffic Engineering I

Monitoring/Measurements I

Wireless Networks I

Routing I

Resource Management and QoS I

Topology and Location Awareness

Caching and Content Management

Optical Networks I

Mobile Ad-Hoc Networks II

Transport Protocols

Monitoring/Measurements II

Mobility/Handoff

Peer-to-Peer

Multimedia

Multicast

Traffic Engineering II

Optical Networks II

Mobile Ad-Hoc Networks III

Wireless Sensor Networks

Resource Management and QoS II

Wireless Networks II

Routing II

Optical Networks III

Poster Session

A Scheme to Provide Proportionally Differentiated End-to-End Packet Delay in Wireless Multi-hop Ad Hoc Networks

Dan Li[1,2] and Peng-Yong Kong[1]

[1] 21 Heng Mui Keng Terrace, Institute for Infocomm Research, 119613 Singapore
[2] Electrical & Computer Engineering Department, National University of Singapore

Abstract. This paper proposes a scheme to provide in a CSMA/CA based multi-hop wireless ad hoc network, a consistent and accurate proportional differentiation in average end-to-end packet delay. The proposed scheme, called PDMED uses a cross-layer approach that requires a distributed scheduler to adapt to the information from a QoS monitor, a route monitor and a channel monitor. Conceptually, the distributed scheduler dynamically adjusts the backoff duration of a flow based on its instantaneous deviation from the maximum average end-to-end packet delay. This is done such that a flow with a larger deviation from the maximum is given a longer backoff duration to give way to transmissions from other flows with smaller deviations. PDMED has been extensively evaluated through random event simulations using OPNET. The results confirm that it is capable of providing a consistent and accurate proportional differentiation, which is otherwise not achievable under various traffic conditions.

Keywords: Proportional Differentiation, Multi-hop, Ad Hoc, End-to-end QoS.

1 Introduction

Wireless multi-hop ad hoc networks can be used to inter-connect various types of sensors without any pre-existing infrastructure. As a result of not relying on any existing infrastructure, multi-hop ad hoc networks have several salient and unique features. First, the network topologies are dynamic and changed often rapidly because of unpredictable and arbitrary movement of nodes. Also, the shared medium nature makes the availability of resource at one node being affected by its contending neighbors. Thus, node interconnectivity and link properties such as capacity and bit error rate cannot be pre-determined. Second, distance between the two ends of a link, obstacles in the environment, externally generated noise and interference caused by other transmissions will make the capacity of a wireless link reduced and apt to be highly variable. Therefore, the wireless link has a bandwidth-constrained and variable capacity. Third, multi-hop ad hoc networks are power-constrained because of lightweight batteries. The limited power supply limits the transmission range, date rate, communication activity and processing speed of the devices. Forth, the multi-hop networks need not have a centralized administration and thus, only local but not global information is available to any node in the network. This implies distributed operations on every node are required.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1–12, 2006.

Given the features presented above, multi-hop ad hoc networks suffer from resource constraints and operation vulnerability and therefore, quality of service (QoS) support in the network becomes a very demanding task [1]. Despite difficult, QoS provisioning in a multi-hop ad hoc network is unavoidable because sensor data do need timely delivery. For example, packets from an image sensor must be delivered real-time so that any illegal intruder can be detected immediately. Also, different types of sensor will require different QoS levels. For instance, packets from a temperature sensor that captures data once every 5 minutes should not be dropped in the presence of an instantaneous resource constrain compared to packets from an image sensor that generates a continuous stream of data.

IEEE 802.11 working group has taken the effort to define a standard mechanism to collectively adjust backoff duration and distributed inter-frame spacing (DIFS) to achieve efficient QoS differentiations [2]. The effort yields CSMA/CA based 802.11e protocol which has been extensively studied in the literatures [3], [4]. From the studies, controlling backoff duration is effective in introducing throughput differentiation while adjusting DIFS duration amplifies the differentiation. The studies also show that 802.11e can provide differentiation when there is a fixed number of active nodes within a radio range in an idealistic channel even though the traffic load is at a saturated level. However, the differentiation is vulnerable to changes in the number of nodes and traffic load. This vulnerability is partly due to the definition of its differentiation where a flow can choose one amongst a small number of service classes (or priorities) that best meet its QoS requirement, based on the assurance that the perceived QoS of higher classes will be better, or at least no worse than that of lower classes. This type of differentiation is called relative differentiation compared to proportional differentiation which offers predictable and controllable differentiations between different service classes [5].

For accurate proportional differentiation in terms of throughput, there exist various methods to map the virtual clock of a fair queuing model into the backoff duration of a CSMA/CA MAC protocol [6], [7]. Unfortunately, all these works can only achieve proportional differentiation locally or globally between two nodes over one hop. With multiple hops, the proportional differentiation should be achieved in an end-to-end manner across all hops but not limited to a concatenation of local proportional differentiations at each hop.

In order to provide QoS across multiple hops, [8] has proposed a distributed packet scheduling algorithm for CSMA/CA based MAC protocols to achieve an accurate transmission order as if in a centralized scheduler that provides QoS differentiation. Based on the desired transmission order, the scheduling algorithm assigns to every packet an appropriate priority. With the priority of a head packet, each node can rank itself against all its neighboring nodes after overhearing their head packets' priorities which are piggybacked on other transmissions. According to the rank, a node will determine its backoff duration to achieve the desired transmission order. Although the algorithm is capable of ensuring an accurate transmission order in a multi-hop setting, it is for packet and not flow. Further, there is no end-to-end performance objective.

For different QoS to different flows across multiple hops, [9] proposes a coordinated multi-hop packet scheduling algorithm that requires some modifications to and cooperations from the CSMA/CA MAC protocol. In [9], the end-to-end QoS requirement of a flow is transformed into an instantaneous priority by the packet scheduling algorithm. Here, a packet that has not been offered sufficient service in the previous hop will be given a higher priority in the future hops and vice versa. The priority of the current and the next packets will be piggybacked onto RTS/CTS and DATA/ACK packets, respectively. Hence, all nodes within a hop know each other's instantaneous priorities and only the node with the highest relative priority will contend for the channel while the other nodes defer their own transmissions. It is the mechanism of adjusting a packet's priority at a hop based on its experience in previous hops that enables end-to-end QoS across multiple hops. The similar service compensation mechanism has been adopted by [10] for the same goal. More aggressively, [10] intends to provide a guarantee in end-to-end packet delay through admission control. Since there is no intuitive way to compute the capacity of a multi-hop ad hoc network, the admission control is done using an admit-then-test method. Specifically, a flow with end-to-end delay requirement is first admitted and then, its impact on the channel idle time is monitored. If the idle time becomes too short as a result of the new flow, another flow that has no end-to-end delay requirement is selected for rejection. Thus, an admitted flow may be dropped. Also, none of these schemes is capable of supporting the end-to-end proportional differentiations which are more controllable and predictable compared to other QoS offerings in a multi-hop ad hoc network.

We have learnt that there are numerous mechanisms across the protocol layers and time scales for QoS delivery in multi-hop ad hoc networks. Among these mechanisms are QoS routing protocols, admission control policies, resource reservation schemes, packet scheduling algorithms, QoS capable MAC protocols, etc. Unfortunately, none of these existing mechanisms is alone capable of providing satisfactory end-to-end proportional differentiations. Logically, a combination of these mechanisms have to work collaboratively to achieve the goal. For example, we may need a packet scheduling algorithm that transforms the QoS requirements into medium access priorities and works with a MAC protocol that provides the multiple priorities. Therefore, this paper contributes in developing a cross-layer scheme to provide proportional differentiation in end-to-end packet delay in wireless multi-hop ad hoc networks.

The remainder of this paper is organized as follows. Section 2 presents in details the proposed cross-layer scheme called Proportionally Differentiated Multi-hop End-to-end Delay (PDMED). The PDMED scheme has been evaluated through random event simulation using OPNET and simulation results are discussed in Section 3. The paper ends with concluding remarks in Section 4.

2 The PDMED Scheme

As illustrated in Fig. 1, PDMED consists of a traffic police, a routing algorithm, a centralized scheduler and a distributed scheduler. These are in turn assisted by a QoS monitor, a route monitor and a channel monitor. In this paper, we assume that all the traffic flows are self-disciplined such that no traffic policing is required. We further assume that all the nodes are not mobile and have a deterministic route quality so that the static shortest path routing protocol can be adopted. We also assume the use of CSMA/CA MAC protocol. This implies the collision avoidance function consists of

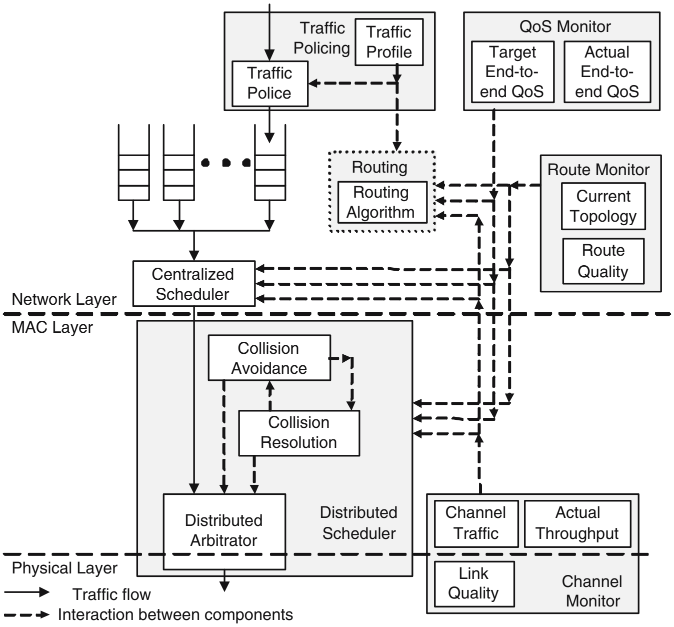


Fig. 1. The cross-layer PDMED scheme

RTS/CTS exchange and carrier sensing. Also, the collision resolution function is based on the paradigm that each flow has its own contention window size. Thus, collisions can be resolved by dynamically adjusting the contention window size based on which the backoff duration of a flow is determined. Let W_i be the contention window size of a flow i. Then, the backoff duration of a flow i, $\varDelta_i$ in terms of number of discrete intervals is decided as follows:

$$\varDelta_i = U[0, W_i - 1], \tag{1}$$

where $U[x, y]$ is a function that generates random integer numbers within the range $[x, y]$. In (1), W_i is adjusted depending on the number of retransmissions, m the current flow i's packet has experienced such that $W_i = 2^m \times W_{min}$, where W_{min} is the minimum contention window size of all flows. While W_i increases with the number of retransmissions, it is upper bounded by W_{max}. The adoption of CSMA/CA also means that the centralized scheduler is implicit. Specifically, with CSMA/CA, only the local flow that has finished first counting down its backoff duration can contend for medium access with the other flows from neighboring nodes.

With the assumptions given above, the task of providing an accurate end-to-end proportional differentiation falls mainly on a distributed scheduler which is presented next. We let the QoS be defined in terms of average end-to-end packet delay. Thus, the target end-to-end QoS of the QoS monitor in Fig. 1 can be written as follows:

$$\frac{d_i(t)}{\phi_i} - \frac{d_j(t)}{\phi_j} = 0;\ \forall\, i, j, t, \tag{2}$$

where ϕ_i is the proportional differentiation parameter and $d_i(t)$ is the actual average end-to-end packet delay for flow i at time t. In practice, $d_i(t)$ must be measured at the destination node of flow i. From the expression above, the target QoS can be interpreted as achieving among all flows an equality in their normalized end-to-end packet delays and the deviation of a flow i from the target QoS at time t can be quantified by $\beta_i(t)$ as follows:

$$\beta_i(t) = \max_{\forall\, j/i} \left\{ \frac{d_j(t)}{\phi_j} \right\} - \frac{d_i(t)}{\phi_i}. \tag{3}$$

From the equation, $\beta_i(t)$ is a positive real number where the smaller its value means closer it is to the QoS target, i.e., $\beta_i(t) = 0$. Thus, $\beta_i(t)$ is also used as the measurement for the actual QoS of flow i at time t.

In order to make $\beta_i(t)$ as close as possible to its target value 0, we propose to dynamically adjust the backoff duration of a flow based on its instantaneous deviation from the equality such that a flow with a relatively smaller $\beta_i(t)$ is given a shorter backoff duration to reduce its end-to-end packet delay. On the other hand, a flow with a relatively larger $\beta_i(t)$ is given a longer backoff duration to give way to transmissions from other flows with a smaller $\beta_i(t)$. However, there is no intuitive best known method to perform the adjustment because of the following two problems: (a) The average end-to-end packet delay, $d_i(t)$ that is measured at the destination node is not readily available to the intermediate nodes and source node of the flow, and (b) The normalized end-to-end packet delay of a flow is only known to the flow itself but the computation of $\beta_i(t)$ requires the normalized delays of other contending flows.

Solving the two problems are the functions of the QoS monitor and channel monitor (refer to Fig. 1), respectively. In the QoS monitor, a backward propagation scheme is proposed so that $d_i(t)/\phi_i$ computed at the destination node will be known by the flow's intermediate and source nodes. According to the backward propagation scheme, when a packet arrives at a flow i's destination node at time t, its average end-to-end delay is updated as follows:

$$d_i(t) = \frac{\tau_i(t) + (n(t) - 1) \times d_i(t')}{n(t)}, \tag{4}$$

where $\tau_i(t)$ is the end-to-end delay of the packet that arrives at time t, $n(t)$ is the total number of packets including the newly arrived one up to time t, and $d_i(t')$ is the previous average packet delay. Through the updating process, the destination node always has the latest value of normalized average end-to-end packet delay, i.e., $d_i(t)/\phi_i$. The latest value together with its respective flow identity will be piggybacked onto the MAC ACK frames that are transmitted in response to each successfully received MAC DATA frame of the flow. At the intermediate nodes, the piggybacked information will be extracted from the received MAC ACK frames and stored locally before being similarly piggybacked onto the upcoming MAC ACK frames of the flow. As such, the actual normalized end-to-end packet delay of each flow can be propagated from the destination node to the source node. We notice that there will be a time lag between the computation of an instantaneous normalized average end-to-end delay and its arrival at the intermediate and source nodes. In practice, the extension of the time lag depends on the number of hops and its impact on the QoS target will be extensively studied through simulation in the next section.

In the channel monitor, a sniffer is proposed to read all the transmitted MAC ACK frames within a broadcast region. With the sniffer, each node can maintain a table containing the identities of all neighboring flows and their respective latest normalized average end-to-end delays. The table is updated each time a MAC ACK frame is received. With the up-to-date table, $\beta_{i,k}(t)$, i.e., the value of $\beta_i(t)$ (refer to (3)) at the k-th hop of flow i can be computed as follows:

$$\beta_{i,k}(t) = \max_{\forall\ j \in \mathcal{I}_{i,k}/i} \left\{ \frac{d_j(t)}{\phi_j} \right\} - \frac{d_i(t)}{\phi_i}, \tag{5}$$

where $\mathcal{I}_{i,k}$ is the set of flow i's neighboring flows at its k-th hop. Based on the computed $\beta_{i,k}(t)$, flow i can rank itself among all its neighboring flows. Specifically, the flow will be given the rank ℓ if its $\beta_{i,k}(t)$ is the ℓ-th highest among all the neighboring flows.

Let $r_{i,k}$ be the rank of flow i at its k-th hop when it has a packet to transmit there but sense a busy channel. In case no ranking can be performed, the default value for $r_{i,k}$ is unity. Also, let $W_{i,k} = 2^{m_{i,k}} \times W_{min}$ be the flow's contention window size at its k-th hop when the packet is making the $m_{i,k}$-th retransmission attempting and $m_{i,k} = 0$ for a fresh packet. Then, instead of using the original CSMA/CA method in (1), the distributed scheduler will decide a flow's backoff duration, $\Delta_{i,k}$ as follows:

$$\Delta_{i,k} = \begin{cases} U[0, W_{min} - 1] + I_{r_{i,k} \geq 2} \times \gamma_{i,k} \times W_{min} \\ \qquad\qquad\qquad\qquad\qquad \text{if } m_{i,k} = 0, \\ U[0, \frac{W_{i,k} - 1}{h_i}] + W_{i,k} \times \left(\frac{h_i - k}{h_i} + r_{i,k} - 1 \right) \\ \qquad\qquad\qquad\qquad\qquad \text{otherwise,} \end{cases} \tag{6}$$

where h_i is the total number of hops for flow i and it is provided to the distributed scheduler by the route monitor in Fig. 1. In (6), the term I_A is an indicator function defined as follows:

$$I_A = \begin{cases} 1 \text{ if } A, \\ 0 \text{ otherwise,} \end{cases} \tag{7}$$

and $\gamma_{i,k}$ is a dynamic control parameter for flow i at its k-th hop. The control parameter has an initial value of unity and it is dynamically adjusted only for a fresh packet at time t based on the actual normalized average end-to-end delay as follows:

$$\gamma_{i,k} = \begin{cases} \gamma'_{i,k} + 1 \text{ if } 0 < \beta_{i,k}(t') < \beta_{i,k}(t) \\ \gamma'_{i,k} - 1 \text{ if } \beta_{i,k}(t) = 0 \text{ and } \gamma_{i,k} > 1 \\ \gamma'_{i,k} \qquad \text{otherwise,} \end{cases} \tag{8}$$

where $\beta_{i,k}(t')$ and $\gamma'_{i,k}$ are the previous values of $\beta_{i,k}(t)$ and $\gamma_{i,k}$, respectively.

Compare (6) to (1), we notice that PDMED scheme gives priority to a flow that experiences excessive normalized average end-to-end delay by allowing a smaller backoff duration. In order to ensure a high responsiveness, $\gamma_{i,k}$ provides an additional degree of freedom when ranking and prioritization alone are not sufficient to quickly bring down a high excessive normalized delay. Also, PDMED gives priority to a retransmitted packet compared to a fresh packet. This is to avoid the situation where multiple packets from a same flow are contending with each other arbitrarily. Among all the retransmitted packets, based on the heuristic disclosed in [11], the packet that is closer to the destination node will be given the priority to transmit so that the overall end-to-end delay can be reduced.

3 Performance Evaluation

We have evaluated PDMED using OPNET. For the purpose of simulation, the general network topology as illustrated in Fig. 2 is used. In the network, there are only two flows, namely Flow 1 (S1-D1) and Flow 2 (S2-D2). From the figure, Flow 1 and Flow 2 have 3 and 2 hops, respectively. For the flows, their differentiation parameters are denoted by ϕ_1 and ϕ_2, respectively.

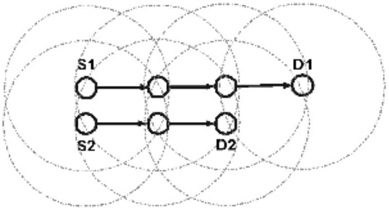


Fig. 2. Network topology

In the simulations, traffic for each flow is generated using a Poisson arrival process with a fixed packet size, L_m and a packet arrival rate, λ. Hence, the packet inter-arrival time is exponentially distributed with mean λ^{-1}. Hereafter, L_m is fixed at 500 bytes unless specified otherwise. In the evaluation, the raw bit rate of communication channel is 1 Mbps. Also, W_{min} and W_{max} are fixed at 16 and 1024 time slots, respectively. Here, the duration of each time slot, $T_{slot} = 50\mu$ second.

First of all, we perform simulations to study the usefulness of the backward propagation scheme adopted by the QoS monitor to inform the nodes of a flow's instantaneous normalized end-to-end delay. Recall that the backward propagation is achieved by piggybacking the latest normalized average delay value onto the MAC ACK frames. We disable the piggybacking in some simulations and compare the results with those of PDMED. The comparison is depicted in Fig. 3 which shows the performance in terms of average end-to-end packet delay. The delay of a packet is the time elapsed since the packet's arrival at the MAC layer of its source node until the packet's subsequent arrival at the MAC layer of its destination node. These packets from their respective traffic sources are queued above but not in the MAC layer to avoid distortion in packet delay at high traffic rate, λ^{-1} when the delays of all flows increase exponentially making any difference in their values not noticeable. In Fig. 3, different ϕ_2/ϕ_1 ratios are achieved by fixing ϕ_1 at 1 while varying ϕ_2. The results show that PDMED can indeed provide a proportional differentiation in average end-to-end packet delay despite that the flows

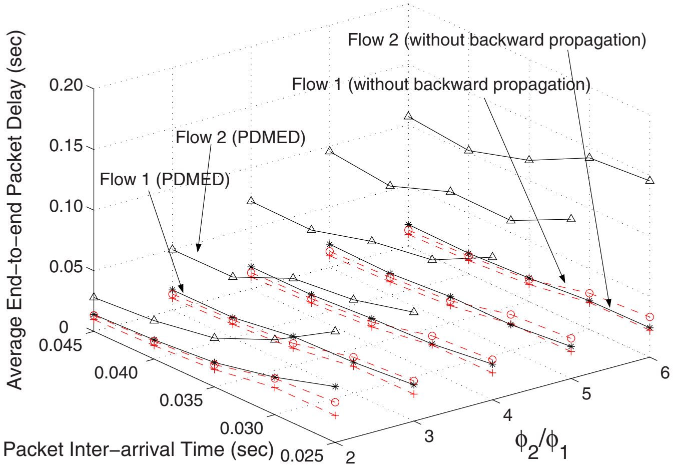


Fig. 3. Average end-to-end packet delay with and without the backward propagation scheme

are going through different numbers of hops. When there is an increase in ϕ_2/ϕ_1, the proportional differentiation is indicated by a rapid increase in Flow 2's end-to-end delay and a slow decrease in Flow 1's end-to-end packet delay although Flow 2 has fewer hops compared to Flow 1. Also, the delays of Flow 1 and Flow 2 increase and keep a fixed differentiation ratio with respect to an decrease in λ^{-1}.

Fig. 3 has confirmed the importance of the backward propagation scheme because, without it, the difference between the two flow's delays is not obvious at various ϕ_2/ϕ_1 ratios. This is further verified in Fig. 4 where the difference between the two flow's normalized average end-to-end packet delay is plotted. Ideally, the difference should

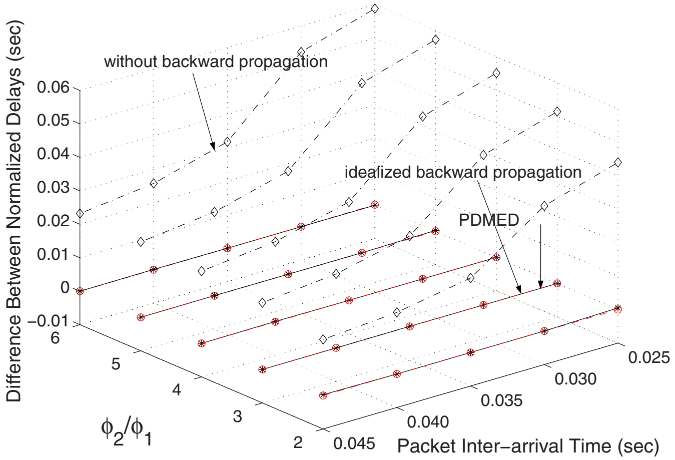


Fig. 4. Difference in normalized end-to-end packet delays with and without the backward propagation scheme

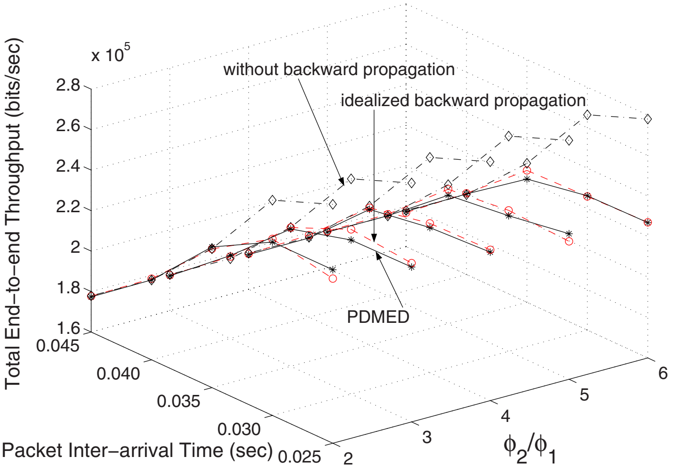


Fig. 5. Total end-to-end throughput as measured at the respective destination nodes, with and without the backward propagation scheme

be zero because, as stated in (2), the performance goal is to achieve equality in the normalized delays. From Fig. 4, PDMED can indeed approximate the performance goal regardless of the traffic rate and ϕ_2/ϕ_1 ratio. On the other hand, the performance goal is not achievable when there is no backward propagation. This happens because, in the absence of the backward propagation, the intermediate nodes do not know the actual end-to-end delay and thus, cannot adjust its backoff duration appropriately to meet the performance goal.

In the evaluation above, the backward propagation scheme is disabled by simply not piggybacking the computed normalized delay on ACK frames. While this leads to a failure in accurate proportional differentiation, there is a noticeable gain in total end-to-end throughput of the two flows as depicted in Fig. 5. This is because, without the instantaneous normalized delay, an intermediate node cannot correctly compute $\beta_{i,k}(t)$ according to (5) and consequently, will not perform the ranking mechanism and adjust $\gamma_{i,k}$ according to (8). Without the ranking and adjustment, $r_{i,k}$ and $\gamma_{i,k}$ stay at their default values of unity. Thus, the backoff duration will always be selected from a range upper bounded by $W_{min} - 1$ compared to a potentially much larger range adjusted by ranking and $\gamma_{i,k}$ according to (6). The smaller backoff duration is the cause of the better end-to-end throughput when there is no backward propagation. In the presence of backward propagation, we treat the reduction in throughput as the cost to pay for the accurate proportional differentiation.

The ranking in PDMED may not always based on the latest instantaneous normalized delay because the backward propagation scheme takes time to distribute the delay across multiple hops after it is computed at the destination node. Specifically, there is always a time lag before the latest normalized delay is available at an intermediate node. Fortunately, this time lag has no significant impact in achieving an accurate proportional differentiation in average end-to-end delay as illustrated in Fig. 4. In the figure, there is no obvious difference in performance when PDMED is equipped with an

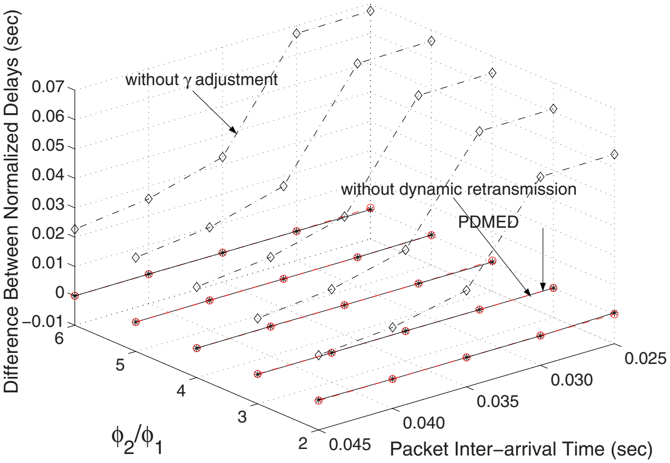


Fig. 6. Difference in normalized end-to-end packet delay with and without the $\gamma_{i,k}$ adjustment and the dynamic retransmission

idealized backward propagation scheme. Compared to the original scheme, the idealized scheme does not require piggybacking of the latest delay on ACK frames. Instead, the simulation program makes the delay known to all the intermediate nodes as soon as it is computed. Without piggybacking, the idealized propagation scheme consumes less bandwidth. However, as shown in Fig. 5, there is no obvious throughput difference between the original and idealized back propagation schemes. This implies the backoff propagation scheme is efficient as it introduces only very small overhead.

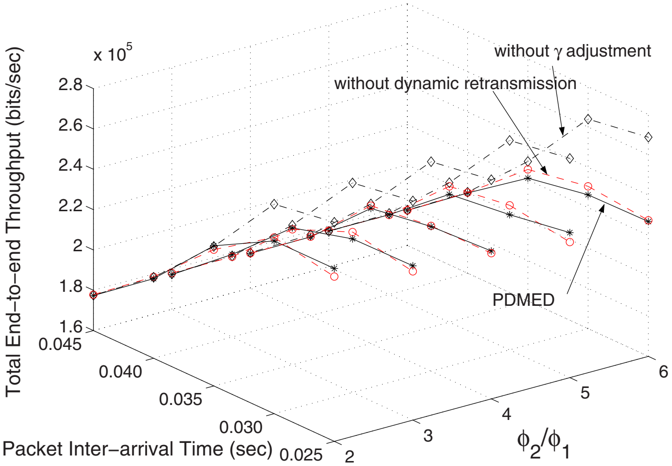
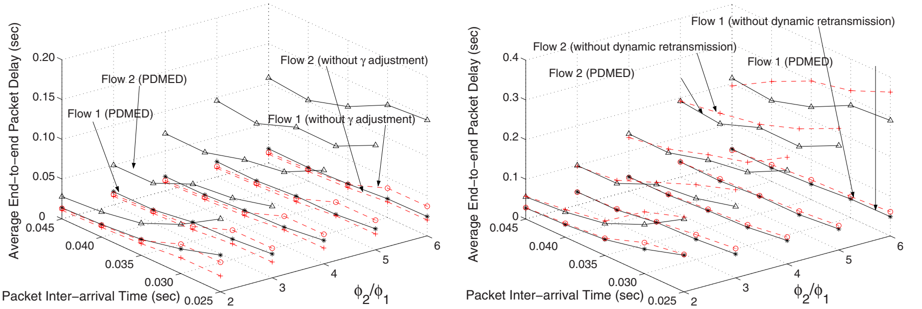


Fig. 7. Total end-to-end throughput as measured at the respective destination node with and without the $\gamma_{i,k}$ adjustment and the dynamic retransmission

Thus far, we have shown the importance and effectiveness of the backward propagation scheme in PDMED. In short, the backward propagation is needed so that intermediate nodes can obtain the instantaneous normalized delay for ranking and $\gamma_{i,k}$ adjustment to achieve an accurate proportional differentiation. Next, we want to show that the ranking itself, without $\gamma_{i,k}$ adjustment is not sufficient. For this purpose, we have repeated the simulations after disabling the adjustment algorithm in (8). Fig. 6 shows the difference between the two flow's normalized average end-to-end packet delay. Compared to PDMED, the larger difference indicates a less accurate proportional differentiation when there is no $\gamma_{i,k}$ adjustment. This means the ranking mechanism alone is not enough in the channel monitor.



(a) With and without the $\gamma_{i,k}$ adjustment (b) With and without the dynamic retransmission

Fig. 8. Average end-to-end packet delay

Although the absence of $\gamma_{i,k}$ adjustment cannot produce an accurate proportional differentiation, it results in a higher total end-to-end throughput as illustrated in Fig. 7. Refer to (6), this is because the backoff duration tends to be smaller when $\gamma_{i,k}$ is not dynamically adjusted but fixed at its initial value of unity. The better throughput without $\gamma_{i,k}$ adjustment also leads to a lower end-to-end packet delay as illustrated in Fig. 8(a). Despite a lower delay, when there is no $\gamma_{i,k}$ adjustment, the difference in delay does not follow the ϕ_2/ϕ_1 ratio and thus does not constitute an accurate proportional differentiation. This is not the case in Fig. 8(b) where we show the impact of the dynamic retransmission scheme in PDMED. As given in (6), a retransmission is indicated by $m_{i,k} > 0$ and the dynamic retransmission scheme gives higher priority to transmissions from a node closer to a flow's destination node. As such, PDMED can deliver a smaller end-to-end delay compared to the case without the dynamic retransmission scheme. The simulations without the retransmission scheme have been performed by simply selecting the backoff duration, i.e., $\Delta_{i,k}$ in (6) from the range $[0, W_{i,k} - 1]$ when capable of reducing end-to-end delay, it does not compromise the accuracy of proportional differentiation and total throughput as illustrated in Fig. 6 and Fig. 7, respectively.

We have evaluated PDMED under various other conditions and benchmarked against IEEE 802.11e using video traces. However, these results are not presented here due to space limitation.

4 Conclusions

Noticing the lack of support in providing proportional differentiation in end-to-end packet delay in a wireless multi-hop ad hoc network, this paper proposes PDMED to do so. PDMED consists of a few mechanisms and monitors which operate across different protocol layers and time scales. PDMED has been extensively evaluated through random event simulation. The results indicate that an accurate and consistent proportional differentiation in end-to-end packet delay which cannot be achieved otherwise, can now be achieved.

References

1. M. S. Corson, “Issues in supporting quality of service in mobile ad hoc networks”, *IFIP 5th Int. Workshop on Quality of Service (IWQOS’97)*, May 1997.
2. M. Benveniste, G. Chesson, M. Hoehen, A. Singla, H. Teunissen and M. Wentink, “EDCF proposed draft text”, *IEEE working document 802.11-01/131-rl*, March 2001.
3. J. Kim and C. Kim, “Performance analysis and evaluation of IEEE 802.11e EDCF”, *Wireless Communications and Mobile Computing*, Vol. 4, No. 1, pp. 55-64, February 2004.
4. B. Li and R. Battiti, “Performance analysis of an enhanced IEEE 802.11 distributed coordination function supporting service differentiation”, *QofIS* LNCS 2811, pp.152-161, 2003.
5. C. Dovrolis, D. Stiliadis and P. Ramanathan, “Proportional differentiated services: Delay differentiation and packet scheduling”, *IEEE/ACM Trans. Networking*, Vol. 10, No. 1, pp. 12-26, February 2002.
6. H. Luo, S. Lu, V. Bharghavan, J. Cheng and G. Zhong, “A packet scheduling approach to QoS support in multi-hop wireless networks”, *Mobile Network and Applications*, Vol. 9, No. 3, pp. 193-206, June 2004.
7. A. K. Somani and J. Zhou, “Achieving fairness in distributed scheduling in wireless ad-hoc networks”, *IEEE IPCCC*, pp. 95-102, April 2003.
8. V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi and E. Knightly, “Ordered packet scheduling in wireless ad hoc networks: Mechanisms and performance analysis”, *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 58-70, 2002.
9. V. Kanodia, C., Li, A. Sabharwal, B. Sadeghi and E. Knightly, “Distributed priority scheduling and medium access in ad hoc networks”, *ACM/Baltzer Wireless Networks*, Vol. 8, No. 5, pp. 455-466, September 2002.
10. Y. Yang and R. Kravets, “Distributed QoS guarantees for realtime traffic in ad hoc networks”, *IEEE SECON*, 2004.
11. B. G. Chun and M. Baker, “Evaluation of packet scheduling algorithms in mobile ad hoc networks”, *Mobile Computing and Communications Review*, Vol. 1, No. 2, June 2002.

Service Differentiation Via Adaptive Gateway Discovery in Ad Hoc Networks Connected to Wired Networks

Mari Carmen Domingo

Telematics Engineering Department, Catalonia University of Technology (UPC),
Av. del Canal Olímpic s/n, 08860 Castelldefels (Barcelona), Spain
Tel.: +34 93 413 70 51
cdomingo@entel.upc.edu

Abstract. A gateway discovery mechanism is necessary to allow wireless nodes in an ad hoc network to route their packets towards a fixed network. Real-time applications have quality of service parameters and require a gateway discovery mechanism that helps them to maintain their requirements. Therefore we propose in this work a new adaptive gateway discovery scheme that adjusts the frequency of the gateway advertisements dynamically. This protocol is able to differentiate services between applications and it cooperates with real-time flows to maintain the desired quality of service. Simulation results investigate the performance of the proposed adaptive scheme and show its effectiveness in comparison with the hybrid mechanism in the simulated scenarios.

1 Introduction

Ad hoc networks [1] have been designed as wireless mobile devices that are able to communicate without having to resort to a pre-existing network infrastructure and without the intervention of a system administrator.

Originally, the investigation was centered in developing isolated and independent ad hoc networks to cooperate in military operations or natural catastrophes like hurricanes. However, this kind of networks is restricted to certain particular environments. Therefore, more recently, the attention has been focused in studying the interaction between ad hoc networks and other types of networks like cellular networks, infrastructure-based WLANs (Wireless Local Area Networks) [2] or wired networks.

The communication between wireless ad hoc networks and infrastructure-based networks is essential to extend Internet beyond its traditional scope, to remote inaccessible areas, making Web services available anytime, anywhere. In addition, ad hoc networks can be seen as an easy way to reduce the congestion in hotspot areas, allowing users to communicate themselves directly without the presence of an access point like in infrastructure-based IEEE 802.11 WLANs; besides, ad hoc networks are able to access Internet via a gateway as well.

The mobile nodes in a wireless ad hoc network must be able to detect available gateways and select one of them if they want to have Internet access.

Real-time applications have special quality of service requirements that must be satisfied to function properly.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 13 – 24, 2006.

We argue that new gateway discovery mechanisms should be designed thinking over the requirements of real-time flows because the selected gateway discovery mechanism will affect the overall performance of the ad hoc network.

Our objective will be to design a new gateway discovery protocol that helps real-time flows to maintain their quality of service parameters. Some different approaches have been developed in literature that propose different gateway discovery schemes, but none of them is related to service differentiation and quality of service improvement for real-time flows. This is the main contribution of this paper.

The paper is organized as follows: Section 2 describes related work about Internet gateway discovery methods. Section 3 remarks the importance of quality of service provision in wireless ad hoc networks. The proposed adaptive gateway discovery scheme is presented in Section 4. Section 5 shows our simulation results and finally Section 6 concludes this paper.

2 Related Work

In order to communicate the ad hoc and the Internet network packets must be transmitted to a gateway as it is illustrated in Fig. 1. This device implements the protocol stack of the ad hoc as well as the fixed network, routing the packets from one network to the other. The protocol stack used by mobile nodes, gateways and Internet nodes is shown in Fig. 2.

The Internet Draft "Global Connectivity for IPv6 Mobile Ad Hoc Networks" [3] describes how to provide Internet connectivity to mobile ad hoc networks modifying an existing routing protocol (like AODV [4] in the example) so that it is able to discover gateways.

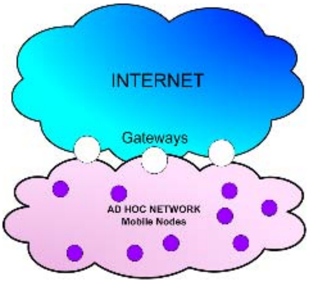


Fig. 1. Interworking scenario

Three main approaches have been developed to detect gateways:

- Proactive gateway discovery [5] [6]: The gateways periodically broadcast advertisement messages that contain information about the global prefix length and the IPv6 address from the gateway. These messages are flooded throughout the entire network. The mobile nodes use this information to autoconfigure a new routable IPv6 address and select the address of one of the gateways as default route. The mobile nodes select the best Internet-gateway by its distance in hops or by other parameters.

- Reactive gateway discovery [7]: A mobile node that wants to send packets towards Internet broadcasts a message to the group of gateways within the ad hoc network. The gateways receive this message and reply to it accordingly. The mobile node selects the gateway which offers the best route towards Internet in terms of number of hops or other parameters.
- Hybrid gateway discovery [8] [9]: This method combines the reactive and proactive approaches; it defines a transmission range where the gateways periodically send advertisement messages and they are propagated around a limited zone (a certain number of hops away from the gateway). A mobile node receiving these messages can obtain information about the global prefix length and the IPv6 address from the gateways carried in this message to discover the global prefix. Afterwards, this mobile node autoconfigures a new routable IPv6 address and selects the address of one of the gateways as default route. The mobile nodes select the gateway that is either closer in terms of number of hops or that is more appropriate because of other parameters. If a mobile node wants Internet connectivity and it is outside the gateways transmission range and the propagation zone of the gateways advertisements, it broadcasts a message to the group of gateways in the ad hoc network. If another mobile node receives this message, it rebroadcasts it until it arrives to a gateway that responds sending back a reply. The mobile node selects the reply of the gateway which offers the best route towards Internet in terms of number of hops or due to other parameters.

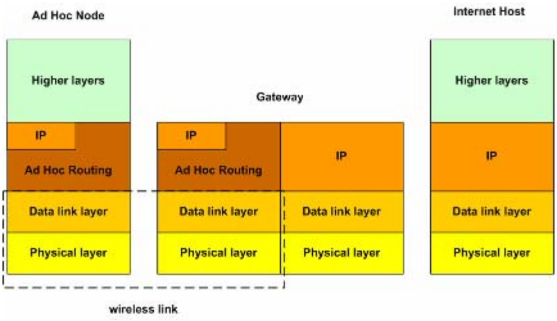


Fig. 2. Protocols architecture

From here on the different approaches that have been proposed in the literature are modifications of the already mentioned gateway discovery strategies.

However, these existing approaches are methods to discover gateways that treat all the traffic in the same way and do not consider differences between real-time and best-effort applications. In the next sections we will remark the importance of providing quality of service to real-time applications in wireless ad hoc networks and of introducing gateway discovery mechanisms that differentiate service levels between best-effort and real-time traffic.

3 Quality of Service Provision in Wireless Ad Hoc Networks

Quality of Service (QoS) can be defined as the ability of the network to offer a required service demanded by a particular application, establishing some type of control over its end-to-end delay, jitter, traffic loss or bandwidth.

It is a very challenging topic to provide QoS in wireless ad hoc networks [10] due to the intrinsic properties of this kind of networks: variable capacity of the links, topologies that change dynamically, etc; furthermore, in wireless networks the packet loss rate and the jitter of the applications are higher in comparison with wired networks due to the existence of fading, interference between neighbouring nodes, etc.

Our objective is to provide QoS to real-time applications in wireless ad hoc networks, differentiating services between real-time and best-effort traffic.

We are interested in studying the performance of multimedia applications in wireless ad hoc networks connected to wired networks. We have selected a specific type of real-time application that implies burstiness and that contains end-to-end delay information: VBR Voice-over-IP (VoIP) [11]. The ITU-T recommends in its standard G.114 that the end-to-end delay should be kept below 150 ms to maintain an acceptable conversation quality [12]. Delays from 150 to 400 ms are acceptable provided that administrators are aware of the impact of quality, and latency larger than 400 ms is unacceptable.

There exists a relation between the QoS provisioning and the gateway discovery method. The hybrid and specially the proactive approaches have a better performance with respect to end-to-end delay, because GWADV messages are sent periodically and not only when it is needed, as in the reactive approach. Thus, real-time applications are able to find a route towards Internet for their traffic sooner. But, on the other hand, if a real-time application has delay problems due to congestion and more GWADV messages are sent, the congestion will be increased and the performance of the delay sensitive applications will be seriously damaged. In the next section we introduce an adaptive gateway discovery approach that has been mainly designed to reduce congestion problems and it helps real-time applications to maintain their QoS parameters even in the presence of excessive traffic.

4 Proposed Adaptive Gateway Discovery Mechanism

The proactive and hybrid approaches have a better performance with respect to end-to-end delay. We have selected as reference model a hybrid gateway discovery mechanism instead of a proactive one because the proactive schemes propagate the GWADVs through the entire network and therefore they introduce more overhead, a circumstance that could seriously damage real-time traffic in the presence of congestion. On the contrary, the GWADVs of the hybrid approach are propagated only a limited number of hops away from the gateway (advertisement zone).

A parameter of the hybrid approach that is directly related to the functioning of this method is the gateway advertisement interval. We have studied the consequences of sending GWADV messages. If mobile nodes receive a GWADV, not all profit in the same way when they receive this gateway message: some nodes benefit and others are harmed with more packets that introduce more congestion and that they don't in fact really need. Consequently, the gateway advertisement interval should be carefully chosen.

We consider a network where best-effort and real-time traffic sources send traffic from the ad-hoc towards the fixed network. We want to provide quality of service, differentiating services between real-time and best-effort applications.

The destination nodes of the real-time traffic in the fixed network periodically monitor the end-to-end delays of these flows. To achieve it, a 'timestamp' or generation time of the packet is introduced in the header of the real-time application protocol (the RTP protocol (Real-time Transport)) and the average end-to-end delay is calculated at the destination node as a time difference. If the end-to-end delay of one or more real-time sources becomes greater than 140 ms, QoS_LOST messages will be sent to the real-time traffic sources that have latency problems to warn them about the situation (see Fig. 3).

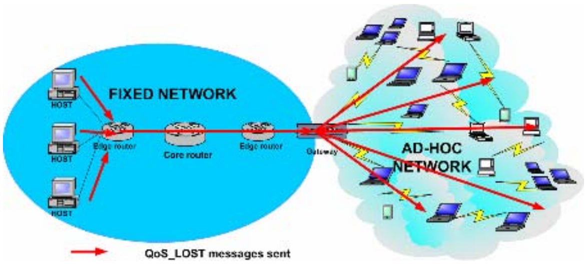


Fig. 3. QoS_LOST messages sent form the fixed towards the ad hoc network

When a node in the ad hoc network receives a QoS_LOST message, it will react executing a QoS mechanism to improve the QoS of its real-time flow; for example the authors in [13] propose a new protocol, named DS-SWAN (Differentiated Services-Stateless Wireless Ad Hoc Networks), to support end-to-end QoS in ad hoc networks connected to one fixed DiffServ domain. DS-SWAN warns nodes in the ad hoc network when congestion is excessive in the ad hoc network for the correct functioning of real-time applications. These nodes react by slowing down best-effort traffic. Simulation results indicate that DS-SWAN significantly improves end-to-end delays of real-time flows without starvation of background traffic.

However, it is not the scope of the paper to suggest which QoS mechanism should be employed to improve the QoS of real-time flows.

We have made the following assumptions aiming to design our adaptive gateway discovery approach:

- Congestion appears in the ad hoc and not in the fixed network. A DiffServ domain in the fixed network may prevent that congestion is introduced in the core routes.
- The destination nodes periodically measure the end-to-end delays of the real-time flows and if these delays are larger than a threshold (140 ms, because the ITU-T recommends to keep these delays under 150 ms and the system needs some reac-

tion time) then QoS_LOST messages will be sent to the source. We are interested in the arrival of QoS_LOST messages to the corresponding gateway that is crossed when these messages travel towards the real-time sources.

- If a route for real-time traffic from the ad hoc towards the fixed network is broken, not only the source node but also the gateway should be warned about the situation with a Route Error (RERR) message. This means that the intermediate node that detects the link failure should send a RERR message not only to the real-time source node but also to the gateway.

We propose a new mechanism where the gateway periodically checks if it has received a RERR coming from a real-time application. In this case, the gateway sends a GWADV message unconditionally because thus this source will be able to find a new route towards the destination sooner with more probability. This does not necessarily mean that this source will sure find a new route by this procedure: It can be possible that the GWADV does not arrive to the source because it is more than TTL hops (advertisement zone) away from the gateway or because there is no route anymore available to reach this source. However, the gateway should not avoid sending this GWADV message because it should try to help a source with routing difficulties.

On the contrary, if the gateway has not received any RERR informing that a real-time source has routing problems, it should do the following:

The gateway should periodically check if it has received QoS_LOST messages during the last *T* seconds from real-time flows having problems to keep their end-to-end delays below 150 ms.

The gateway should calculate:

$$\alpha(t) = \frac{P}{F}, \tag{1}$$

where P = number of real-time sources having end-to-end latency problems and F = total number of real-time sources using that gateway.

We consider a threshold γ, where $0 \leq \gamma \leq 1$. It is fulfilled:

If $\alpha(t) \geq \gamma$, no GWADV messages should be sent by the gateway to the ad hoc network, because if real-time flows have QoS problems due to excessive congestion, it is not recommended to introduce more traffic overload in the network with these messages.

On the contrary, if $\alpha(t) < \gamma$, GWADV messages should normally be sent towards the ad hoc network.

This functioning method serves the purpose that real-time sources do not increase their end-to-end latency problems if congestion is excessive.

The value of γ should be carefully chosen by operators according to their own needs. In the extreme case γ=1, the gateway discovery approach is equivalent to the hybrid scheme.

On the other hand, it is important to think about the role of background best-effort traffic sources. They don't have an active role in the decisions taken by the gateway to help real-time sources to discover the best route for sending their packets towards Internet. Sometimes they will take advantage and sometimes they will be at disadvantage due to the intrinsic functioning of this gateway discovery approach; what is a fact is that this mechanism has not been designed thinking on them although they will try to profit from the situation if it is possible.

The functioning of this adaptive scheme is illustrated in Fig. 4.

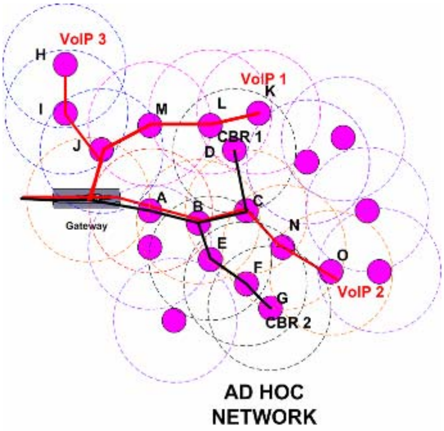


Fig. 4. Example network

It shows an example of an ad hoc network where three VoIP real-time and two CBR best-effort flows have been established to send packets towards Internet through the gateway. If we consider that the VoIP flows VoIP1 and VoIP3 have problems to keep their end-to-end delays under 150 ms, QoS_LOST messages will be sent to these VoIP sources in the ad hoc network through the gateway to warn them about the situation. The gateway takes advantage of this information and it periodically calculates the percentage of VoIP sources that route their packets towards Internet through it and that have end-to-end delay problems. In our example this percentage is $\alpha(t) = 2/3$. If the threshold for latency problems is set to be $\gamma = 0.5$, then it follows that $\alpha(t) \geq \gamma$, which means that no GWADV messages should be sent by the gateway to the ad hoc network, because the number of VoIP sources having delay problems due to excessive congestion is larger than the threshold and this means that the network should not be overloaded with more traffic if it is not strictly necessary. If afterwards one of the VoIP sources solves its QoS problems, the gateway will calculate a new percentage $\alpha(t) = 1/3$. Now GWADV messages should be sent towards the ad hoc network because $\alpha(t) < \gamma$. The advertisement messages will be propagated around a limited zone (a certain number of hops away from the gateway); in this case we consider an advertisement zone of TTL = 4 hops. This means that the gateway advertisement messages will be received by the sources VoIP1 (route gateway-J-M-L-K), VoIP3 (route gateway-J-I-H) and CBR1 (route gateway-A-B-C-D). The other sources will not receive the GWADV messages because they are more than 4 hops away from the gateway and they would have to do a route discovery in the case that the route towards the gateway breaks.

5 Simulations

We have run simulations with the NS-2 tool [14] to investigate the performance of our proposed approach. The system framework is shown in Fig. 5. A scenario where an ad-hoc network is connected via two gateways to a fixed IP network has been selected. The chosen scenario consists of 20 mobile nodes, 2 gateways, 3 fixed routers and 3 corresponding hosts.

The mobile nodes are uniformly distributed in a rectangular region of 1000 m by 500 m. The gateways are placed with x, y coordinates (150,250) and (850,250). Each mobile node selects a random destination within the area and moves toward it at a velocity uniformly distributed between 0 and 3 m/s. Upon reaching the destination, the node pauses for a pause time, selects another destination and repeats the process. Five different pause times have been used: 0, 20, 50, 125 and 200 seconds. The dynamic routing algorithm is AODV [4] and the wireless links are IEEE 802.11b.

Background traffic is generated by 6 of the mobile hosts, while VBR VoIP traffic is generated by 15 of the mobile hosts. The destinations of each of the background and VoIP flows are chosen randomly among the three hosts in the wired network.

We assume that best-effort CBR background traffic and real-time VBR VoIP traffic are transmitted. We have proposed CBR as background traffic instead of TCP. The reason is that TCP performs poorly in an ad-hoc network because packets that are lost due to link failure and route changes trigger TCP's congestion avoidance mechanisms [15]. On the contrary, many authors [16] use CBR as background traffic successfully.

The VBR mode is used for VoIP traffic. We employ a silence suppression technique in voice codecs so that no packets are generated in the silence period. For the voice calls, we use the ITU G. 726 or "adaptive differential pulse code modulation (ADPCM)" codec. The VoIP traffic is modelled as a source with exponentially distributed on and off periods with 1.004 s and 3.587 s average each and two frames (20 ms audio sample each frame) are carried in each packet (80 + 80 bytes payload). Frames are generated during the on period every 20 ms with size 80 bytes and at a constant bit rate of 32 Kbps without any compression. VoIP is established over real-time transport protocol (RTP), which uses UDP/IP between RTP and link layer protocols. Packets have a constant size and are generated at a constant inter-arrival

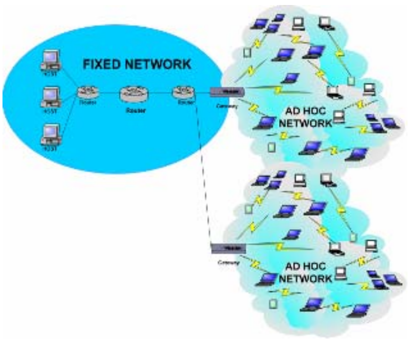


Fig. 5. Simulation framework

time during the on period. The VoIP connections are activated at a starting time chosen from a uniform distribution in [10 s, 15 s].

Background traffic is Constant Bit Rate (CBR) with a rate of 48 Kbit/s and a packet size of 120 bytes. To avoid synchronization, the CBR flows have different starting times that have been randomly chosen.

We have run simulations to assess the following three performance measures:

- Average end-to-end delay for real-time (VoIP) traffic: Defined as the time it takes for data packets to arrive from the source node to the destination node.
- Packet delivery ratio: Defined as the number of real-time (VoIP) packets successfully delivered over the number of real-time (VoIP) packets generated by the sources.
- Routing overhead: Defined as the amount of control packets (for gateway discovery and routing) divided by the sum of the control packets plus the data packets.

We have evaluated and compared the performance of the hybrid approach "Hybrid scheme" with our proposed adaptive scheme discussed in Section 4: "Proposed adaptive scheme". In both approaches a TTL=2 hops is used as advertisement zone.

Fig. 6 shows the average end-to-end delay for VoIP traffic. We can observe that in both schemes the end-to-end delays for VoIP traffic are increased with smaller pause times, because when the pause time is very low the routes of the existing flows break down frequently and the routing protocol continuously does new route discoveries processes that increase the latency. On the contrary, when the pause time is higher, the average link duration is increased as well as the duration of the routes.

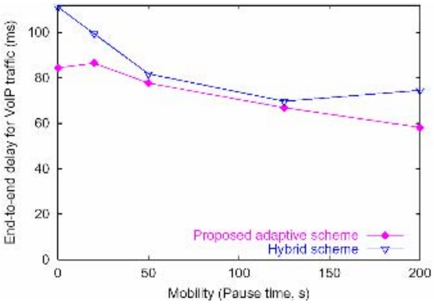


Fig. 6. Average end-to-end delay for VoIP traffic

We notice that the average end-to-end delays for VoIP sources are lower when our proposed adaptive scheme is used, because less GWADVs messages are sent in congestion conditions. The gateway periodically checks if it has received QoS_messages associated with VoIP sources having end-to-end delay problems. If the percentage of VoIP traffic sources having latency problems exceeds a predefined threshold (in this case this threshold is set to $\gamma = 0.15$), no GWADV messages are sent by the gateway. Therefore, no more traffic overload is introduced in a congested network and as a consequence the latency of VoIP flows is diminished. When the pause time is 0 or 200 seconds (extreme cases), the routes break down frequently due to continuous mobility or because of the isolation of certain nodes that don't find neighbours to act as intermediate nodes for their packets and as a result new route discoveries have to be done frequently and congestion is increased more; hence real-time sources have very often delay problems. If a scheme like the proposed adaptive scheme is used, the reduction of congestion is very effective in comparison with the hybrid scheme. A similar trend is observed regarding the jitter for VoIP traffic.

The packet delivery ratio for VoIP packets is illustrated in Fig. 7.

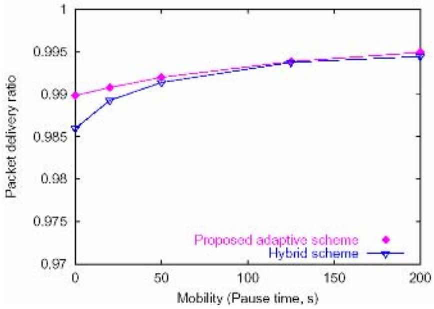


Fig. 7. Packet delivery ratio for VoIP packets

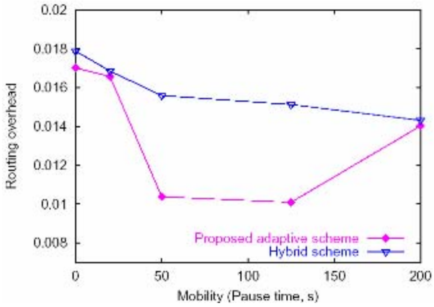


Fig. 8. Overhead of control packets

As we can appreciate from the figure, the higher the mobility of the nodes, the best performs our approach in comparison with the other. There is a trade-off between the signaling overhead and the proactivity of the protocol in both approaches; however, when mobility is high and due to the frequently routes breakage, in the hybrid protocol signaling overhead is higher in comparison with the proposed adaptive approach. Consequently, more congestion is introduced and as a result, the packet delivery ratio is decreased. The differences in packet delivery ratio are lesser as mobility is decreased.

The overhead of control packets is depicted in Fig. 8. We can see that the proposed approach has a lower overhead than the hybrid approach. The differences in overhead are lower in the extreme cases (pause time 0, 20 and 200) because in the proposed approach less GWADV messages are sent by the gateways but, on the other hand, more sources start a route discovery mechanism as in a reactive approach when they need to find a gateway and moreover if a RERR message arrives to a gateway it sends GWADV messages unconditionally so that the overhead is increased. However, the routing overhead is a metric that shows how much network resources does the protocol need to do its work and we can appreciate that the number of network resources is much lower using the proposed mechanism.

Besides, we have done more simulations that show that with the proposed scheme there is no starvation of best-effort traffic.

6 Conclusions and Future Work

We have proposed a new adaptive gateway discovery protocol that differentiates services between applications through gateway selection. The scheme introduced outperforms the hybrid scheme in terms of average end-to-end delays and jitter for real-time flows, packet delivery ratio and routing overhead. What is more, using this scheme there is no starvation of best-effort traffic.

As future work we are planning to do more simulations that evaluate the effectiveness of our gateway discovery scheme when scalability is introduced (with respect to the number of real-time sources, the number of gateways, the traffic load, etc). Besides, we think it could be interesting to analyze the performance of the proposed protocol in a real network.

Acknowledgment

This work was partially supported by the "Ministerio de Ciencia y Tecnología" of Spain under the project TIC2003-08129-C02, which is partially funded by FEDER.

References

[1] S. Basagni, M. Conti, S. Giordando, and I. Stojmenovic, “Mobile Ad Hoc Networking”, IEEE Press & Wiley Inter-Science, 2004.

[2] D. Cavalcanti, C. Cordeiro, D. Agrawal, B. Xie and A. Kumar, “Issues in Integrating Cellular Networks, WLANs, and MANETs: A Futuristic Heterogeneous Wireless Network”, IEEE Wireless Communications Magazine, Vol. 12, No. 3, pp. 30-4, April 2005.

[3] R. Wakikawa, J. T. Malinen, C. E. Perkins, A. Nilsson, and A. J. Tuominen, “Global connectivity for IPv6 mobile ad-hoc networks”, Internet Engineering Task Force, Internet Draft (Work in Progress), July 2002.

[4] C. E. Perkins, E. M. Belding-Royer, and I. Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing", IETF Internet draft, draft-perkins-manet-aodvbis-00.txt, Oct 2003.

[5] Y. Sun, E. M. Belding-Royer and C. E. Perkins, “Internet Connectivity for Ad Hoc Mobile Networks”, International Journal of Wireless Information Networks, vol. 9, issue 2, 2002 , pp. 75-88.

[6] C. Jelger, T. Noel and A. Frey, “Gateway and address autoconfiguration for IPv6 adhoc networks”, Internet-Draft, draft-jelger-manet-gateway-autoconf-v6-02.txt, April 2004.

[7] J. Broch, D. Maltz and D. Johnson, “Supporting Hierarchy and Heterogeneous Interfaces in Multi-hop Wireless Ad Hoc Networks”, in Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms and Networks, June 23-25, Perth, Australia, pp. 370-375.

[8] P. Ratanchandani and R. Kravets, “A Hybrid Approach to Internet Connectivity for Mobile Ad Hoc Networks”, in Proc of the IEEE WCNC 2003, Vol. 3, pp. 1522-1527.

[9] J. Lee, D. Kim, J. J. Garcia-Luna-Aceves, Y. Choi, J. Choi and S. Nam, “Hybrid Gateway Advertisement Scheme for Connecting Mobile Ad Hoc Networks to the Internet”, in Proc. of the 57th IEEE VTC 2003, Vol. 1, Jeju, Korea, April 2003, pp. 191-195.

[10] K. Wu and J. Harms, "QoS Support in Mobile Ad-hoc Networks," Crossing Boundaries-the GSA Journal of University of Alberta, Vol. 1, No. 1, Nov. 2001, pp.92- 106.

[11] D. Chen, S. Garg, M. Kappes and K.S. Trivedi, “Supporting VBR Traffic in IEEE 802.11 WLAN in PCF Mode,” in Proc. OPNETWORK’02, Washington D.C., Aug. 2002.

[12] ITU-T Recommendation G.114, “One way transmission time", May 2000.

[13] M C. Domingo and D. Remondo, “An Interaction Model and Routing Scheme for QoS Support in Ad Hoc Networks Connected to Fixed Networks”, vol. 3266 of Lecture Notes in Computer Science, Berlin, 2004, Springer Verlag, pp. 74-83, ISBN 3-540-23238-9.

[14] NS-2: Network Simulator, http://www.isi.edu/nsnam/ns.

[15] A. Jain, A. Pruthi, R.C. Thakur, and M.P.S. Bhatia, “TCP analysis over wireless mobile ad hoc networks”, IEEE Personal Wireless Communications, New Delhi, India, Dec. 2002.

[16] P.B. Velloso, M. G. Rubinstein and M. B. Duarte, “Analyzing Voice Transmission Capacity on Ad Hoc Networks”, International Conference on Communications Technology - ICCT 2003, Beijing, China, April 2003.

Stability-Throughput Tradeoff and Routing in Multi-hop Wireless Ad-Hoc Networks

Arzad Alam Kherani[1], Rachid El Azouzi[2], and Eitan Altman[3]

[1] Department of computer science and engineering,
Indian Institute of technology Delhi, New Delhi, India
[2] LIA/CERI, Univesité d'Avignon, Agroparc, BP 1228, 84911, Avignon, France
[3] NRIA, 2004 Route des Lucioles, 06902 Sophia Antipolis Cedex, France

Abstract. We study the throughput of multi-hop routes and stability of forwarding queues in a wireless Ad-Hoc network with random access channel. We focuse on wireless with stationary nodes, such as community wireless networks. Our main result is charactrerization of stability condition and the end-to-end throughput using the balance. We also investigate the impact of routing on end-to-end throughput and stability of intermediate nodes. We find that i) as long as the intermediate queues in the network are stable, the end-to-end throughput of a connection does not depend on the load on the intermediate nodes, ii) we showed that if the weight of a link originating from a node is set to the number of neighbors of this node, then shortest path routing maximizes the minimum probability of end-to-end packet delivery in a network of weighted fair queues with coupled servers. Numerical results are given and support the results of the analysis.

1 Introduction

Consider a set of *static* devices spread over some region. Each of these devices is a wireless transceiver that transmits and receives at a single frequency band which is common to all the devices. Over time, some of these devices collect/generate information to be sent to some other device(s). Owing to the limited battery power that these devices are allowed to use, a device may not be able to directly communicate (transmit) with far away nodes. 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. This intermediate device then uses the same procedure so that the information finally reaches its destination[1].

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

[1] We will see later that it is also possible that some of the information is lost before reaching the destination device.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 25–40, 2006.

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 *weights* 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 many attempts have been made to (i) identify the characteristics of the quantities that one would like to associate to a *link* as its weight, and in particular (ii) to understand the resulting network performance and resource utilization (in particular, the stability region and the achievable throughput regions). Some simple heuristics have been frequently reported to improve performance of applications in mobile ad-hoc networks (see [9] 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). We assume that time is slotted into fixed length time frames. In any slot, a node having a packet to be transmitted to one of its neighboring devices 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 successful. We assume throughout that there is some mechanism that notifies the sender of success or failure of its transmissions. For example, the sources get the feedback on whether there was zero, one or more transmissions (collision) during the time slot.

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 *forwarded*.

Clearly, a device needs to have some scheduling policy to decide on which of these types it wants to transmit, given that it decided to transmit. Having a first come first served scheduling is one simple option. Yet another option is to have two separate queues for these two types and do a weighted fair queueing (WFQ) for these two queues. In this paper we consider the second option.

Working with the above mentioned system model, we study the impact of routing, channel access rates and weights of the weighted fair queueing on throughput, stability and fairness properties of the network.

It is worth mentioning that the above scenario may also be studied in the perspective of game theory in which case the nodes are assumed to be rational and need some incentive to forward data from other nodes. Typically in such scenario, a Nash equilibrium determines the operating point (routing, channel access rates and WFQ weights). Thus, the results of this paper may be helpful in comparing various operating points based on criteria of throughput, stability and fairness in the cases where Nash equilibrium is not unique.

Our main result is concerned with the stability of the forwarding queues at the devices. 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 devices. Further, the weights of the WFQs play a role only in determining the tradeoff between the power allocated for forwarding and the delay of the forwarded traffic. The end-to-end throughput achieved by the nodes are independent of the choice of the WFQ weight.

Remark. Most of the studies on random access in wireless networks assume that the sources always have data to send. This then is expected to give the *saturation performance*, which may be the throughput or probability of collision or some similar quantity of interest.

Related literature. Wireless network stability has attracted much interest. Among the most studied stability problems are scheduling [11, 12] as well as for the Aloha protocol [1, 10, 14]. Tassiulas and Ephremides [11] 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 [3] 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 [3], Huang and Bensaou [7] or Tassiulas and Sarkar[13]). Our setting is different than the mentionned ones in the following: the number of retransnmitions, (which is one of the parameters that we optimize) is finite, and therefore in our setting, the output and the input need not be the same.

2 Network Model

In this section, we describe the working of the network in detail and introduce various quantities that determine the overall performance. We provide also the assumptions underlying this study and introduce appropriate notations.

2.1 Assumptions and Definition

Consider a wireless ad-hoc network consisting of N nodes (we allow $N = \infty$ to study some simple symmetric cases without boundary effects). When N is finite, we number the nodes using integers $1, \ldots, N$. We assume a simple channel model:

- A node can decode a transmission successfully iff there is no other interfering transmission.
- Assume that all nodes share the frequency band, and time is assumed to be divided into fixed length slots. - Queues at Nodes , has two queues associated with it: one queue (denoted Q_i) contains the packets that originate at

node i and the other queue (denoted F_i) contains packets that node i has received from one of its neighbors and has to be transmitted (forwarded) to another neighbor. If node i decides to transmit when both the queues Q_i and F_i are nonempty, it implements a weighted fair queue, i.e., node i sends a packet from queue F_i with probability f_i and sends a packet from Q_i with probability $1 - f_i$. If only one of these queues is non-empty, the node selects packet from this non-empty queue to transmit. 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.

- Arrival of data packets at a source node: We assume that the queue Q_i is always nonempty for nodes which are sources of data; this is the case, for example, when the nodes are sensors and they make new measurements as soon as the older ones are transmitted. This kind of models with assumption of *saturated* nodes are intended to provide insights into the performance of the system and also helps study effects of various parameters.

This model allows us to define a neighborhood relation between any two nodes: node i is neighbor of node j if node i can receive transmission from node j in absence of any other transmission. We use the function $A(\cdot,\cdot) : [1,N] \times [1,N] \rightarrow \{0,1\}$ to denote the neighborhood relation: $A(i,j) = 1$ iff i is neighbor of j. We assume that the (binary) neighborhood relation is symmetric, i.e. $A(i,j) = A(j,i)$. Let $\mathcal{N}(i)$ denote the nodes which are neighbors of node i, i.e., $\mathcal{N}(i) = \{j : A(j,i) = 1\}$.

2.2 Channel Access Mechanism

As mentioned before, the time is assumed to be divided into fixed length slots. We assume that the packet length (or, transmission schedule length) is fixed throughout system operation. If node i has a packet waiting to be transmitted in either Q_i or F_i, then node i will attempt a transmission in a slot with some probability P_i, i.e., even when the node is ready to transmit, it may transmit or not in the slot, depending on the collision avoidance and resolution schemes being used, as well as the channel's current state. If the transmission is meant for some node $j \in \mathcal{N}(i)$, then the transmission from node i to j is successful iff none of the nodes in the set $j \cup \mathcal{N}(j) \backslash i$ transmits. This mechanism models the CSMA/CA random channel access mechanism which forms the basis of slotted ALOHA systems. Here we restrict ourselves to a fixed probability of channel access P_i for node i, i.e., the transmission probability does not account for the exponential backoff mechanism sometimes used in CSMA/CA channel access mechanisms in order to reduce the probability of successive collision of a packet. *To avoid pathological cases, in this paper we will assume that* $0 < P_i < 1, \ \forall i$.

2.3 Routing and Packet Loss

Routing is an essential task of transferring packets of information from the sources to the destination. We consider static source routing, i.e., when the source node sends a packet, it appends the information of route that the packet has to follow in the network. This information can be obtained, for example, by a proactive protocol as OLSR[4] and WRP[8]. These protocols contain routing table information by broadcasting control packet and attempt to maintain at all times up-to-date routing information from each node to every other node. By a route from node i to j we mean an ordered sequence of nodes which will forward packets that originate at i and have node j as their destination. By ordered set we mean here that the two successive elements in the set representing a route must be neighbors of each other. Also, the first element of this set is the source and the last element is the destination. We use the notation $R_{i,j}$ to denote the route from node i to j with the nodes i and j removed, i.e., $R_{i,j}$ denotes the ordered set of *intermediate* nodes on route from node i to j. Also, $R_{i,j,k}$ is used to denote the (ordered) subset of all nodes that occur not after node k in the set $R_{i,j}$. Note that we are assuming that all the packets from i to j follow the same route, i.e., there is no probabilistic routing at a packet level.

We assume that all the queues in the network are large enough so that there is no packet *drop* due to buffer overflow. The only source of packet losses that we consider are those arising from excessive number of repeated collisions of a transmitted packet. Specifically, if node i is sending packet on route from node s to d, then if this packet has been attempted transmission $K_{i,s,d}$ number of times by node i and has suffered a collision every time, the packet is dropped. Note that here we allow for $s = i$.

3 Stability Properties of the Forwarding Queues: The Saturated Node Case

First objective of our analysis is to study the effect of the choice of the parameters of the schemes mentioned above (P_i's, $P_{i,j}$'s, routing and the parameter $K_{i,s,d}$'s) on the network performance, i.e., we derive the protocol's performances based on the heavy traffic, i.e., a node always has a packet in its buffer to be sent.

For a given routing, let π_i denote the probability that node i has packets to be forwarded, $\pi_{i,s,d}$ is the probability that queue F_i is nonempty and the packet in the first position in the queue F_i is from the route s to d and n_i is the number of neighboring nodes of node i.

3.1 The Rate Balance Equations

We fix a node i and look at its forwarding queue, F_i. It is clear that if this queue is stable then the output rate from this queue is equal to the input rate into the queue. Only issue to be resolved here is to properly define the term *output rate*. This is because, owing to a bound $K_{i,s,d}$ on the number of attempts for transmission of any packet, not all the packets arriving to F_i may be successfully transmitted. Hence, the output rate is defined as the rate at which packets from queue F_i are either successfully forwarded or are dropped owing to excessive number of collisions. Next we derive the expressions for input and output rates for queue F_i from first principles.

We start by obtaining the *detailed balance equations*, i.e., the fact that if the queue F_i is stable, then the input rate on any route using queue F_i is equal to the output rate from queue F_i on that route.

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

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

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

Lemma 1. *For any node i, s and d such that $P_{s,d} > 0$ and $i \in R_{s,d}$, the long term average rate of departure of packets from node i on route from node s to node d is $\frac{\pi_{i,s,d} P_i f_i}{L_{i,s,d}}$.*

Proof: see the full version of our paper [2]

Lemma 2. *For any fixed choice of nodes i, s and d such that $P_{s,d} > 0$ and $i \in R_{s,d}$, the long term average rate of arrival of packets into F_i for $R_{s,d}$ is*

$$P_s(1 - \pi_s f_s) P_{s,d} P_{s,s,d} \Pi_{k \in R_{i,s,d} \setminus i} \sum_{l=1}^{K_{k,s,d}} (1 - P_{k,s,d})^{l-1} P_{k,s,d}.$$

Proof: See the full version of our paper [2]

Proposition 1. *In the steady state, if all the queues in the network are stable, then for each i, s and d such that $i \in R_{s,d}$,*

$$\begin{aligned}\frac{\pi_{i,s,d} P_i f_i}{L_{i,s,d}} &= P_s P_{s,d}(1 - \pi_s f_s) P_{s,s,d} \Pi_{k \in R_{i,s,d} \setminus i} \sum_{l=1}^{K_{k,s,d}} (1 - P_{k,s,d})^{l-1} P_{k,s,d} \\ &= P_s P_{s,d}(1 - \pi_s f_s) P_{s,s,d} \Pi_{k \in R_{i,s,d} \setminus i}(1 - (1 - P_{k,s,d})^{K_{k,s,d}})\end{aligned}$$

Proof: If the queue F_i is stable, then the rate of arrival of packets on route $R_{s,d}$ into the queue is same as the rate at which the packets are removed from the queue (either successfully forwarded or dropped because of excessive collisions). •

Let

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

and $y_i = 1 - \pi_i f_i$. Note that $w_{s,i}$ *are independent of* $f_j, 1 \le j \le N$ *and depend only on the probabilities* P_j, $P_{s,d}$ *and the routing.*

Theorem 1. *In the steady state, if all the queues in the network are stable, then for each i, s and d such that $i \in R_{s,d}$,*

$$1 - y_i = \sum_s y_s w_{s,i}.$$

Proof: Summing both the sides of the expression in Proposition 1 for all $s, d : i \in R_{s,d}$, we get the *global* rate balance equation for queue F_i. •

The system of equations in Theorem 1 can be written in matrix form as

$$\underline{y}(I + W) = \underline{1}, \tag{1}$$

where W is an $N \times N$ matrix whose $(s,i)^{th}$ entry is $w_{s,i}$ and $\underline{y}$ is an $N-$dimensional row vector.

The relation of Equation 1 has many interesting interpretations/implications. Some of these are:

- *The Effect of f_i*: At the heart of all the following points is the observation that the quantity $y_i = 1 - \pi_i f_i$ is *independent* of the choice of f_j, $1 \leq j \leq N$. It only depends on the routing and the value of P_j.
- *Stability*: 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 *can be made stable by appropriate choice of f_is*. Now, since y_i is determined only by routing and the probabilities $P_j s$ and $P_{s,d}$, we can then *choose* 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.
- *Throughput*: We see that the long term rate at which node s can serve its own data meant for destination d is $P_{s,d} P(1 - \pi_s f_s) = P_{s,d} P y_s$ which is *independent of f_s*. Also, the throughput, i.e., the rate at which data from node s reaches their destination d. This quantity 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 $f_j, 1 \leq j \leq N$.
- *Choice of 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.

- *Energy-Delay Tradeoff*: 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 *throughput-delay* tradeoff that can be obtained by changing the forwarding probabilities. However, we do obtain an *energy-delay tradeoff* because now, for a given *stable routing*, we need to find value of f_i which will determine π_i. Clearly, f_i represents the forwarding energy and π_i gives a measure of the delay.
- *Throughput-Stability Tradeoff*: In the present case, we can tradeoff throughput with stability and not directly with the delay. This is achieved by controlling the routing. This point will be further dealt with in Section 4.
- *Per-route behavior*: Note that the above observations are based on the global rate balance equation for forwarding queue F_i of node i. Similar observations can be made when considering the detailed balance equation for queue F_i for some fixed source destination pair s, d such that $i \in R_{s,d}$.

3.2 Balance Equations Under Unlimited Attempts :$K_{i,s,d} \equiv \infty$

In this subsection, we consider an extreme case in which a node attempts forwarding of a packet until the transmission is successful. This case provides some further important observations while keeping the expressions simple. The detailed balance equation for queue F_i on route from node s to node d is

$$\pi_{i,s,d} f_i P_i P_{i,s,d} = P_{s,d} P_s (1 - \pi_s f_s) P_{s,s,d}.$$

By assuming that all nodes have same channel access rate $P_i = P$, $\forall i$, we have

$$\pi_i f_i = \sum_{s,d: i \in R_{s,d}} \frac{P_{s,d}(1 - \pi_s f_s) P_{s,s,d}}{P_{i,s,d}}.$$

Hence, introducing the transformation $y_i = 1 - \pi_i f_i$, we see that the above set of rate balance equations can be written in matrix form as

$$\underline{y}(I + W_\infty) = \underline{1}$$

where W_∞ is a matrix with its $(s,i)^{th}$ entry being $w_{s,i} = \sum_{d: i \in R_{s,d}} \frac{P_{s,d} P_{s,s,d}}{P_{i,s,d}}$.

Observe that if a source has at most one destination, i.e, $P_{s,d} \in \{0, 1\}$, and if the number of neighbor is same for all the nodes so that $P_{i,s,d} = P_{s,s,d}$, then the rate balance equations become

$$y_i + \sum_{s: i \in R_s} y_s = 1.$$

The above relation has many interesting interpretations/implications. Some of these are:

Stability : if a node s' which is also a source for some destination d' does not forward packets of any other connection, i.e., if $\pi_{s'} = 0$ then for any $i \in R_{s',d'}$, the rate balance equation is

$$\pi_i f_i = \sum_{s,d:i \in R_{s,d}, s \neq s'} (1 - \pi_s f_s) + 1,$$

implying that the forwarding queues of all the nodes in $R_{s',d'}$ are unstable since the above requirement requires $\pi_i \geq 1$ as f_i is bounded by 1. This implies that a *necessary condition for the forwarding queues in the network to be stable is that all the sources must also forward data.* This can have serious implications in case of ad-hoc networks. There is also an advantage of the above result as it reduces the allowed set of routes and thus makes the search for the optimal route easier. From the above rate balance equation it follows that, for a given P and P_f, the stability of the forwarding queue of node i depends in *an inverse manner* on the stability of the forwarding queues of the source nodes of the routes that pass through node i. Precisely, observe that the value of π_i increases with a decrease in value of π_s. This implies that if the routing is such that node i carries traffic of a source s which does not forward any route's packet, i.e., $\pi_s = 0$, then the value of π_i is more as compared to the case where, keeping everything else fixed, now node s forwards traffic from some route.

4 Stability of Forwarding Queues and Routing

In the following we will restrict ourselves to symmetric networks, i.e., we will assume that $P_i = P, \ \forall i$ and $f_i = P_f, \ \forall i$. However, we allow for general source-destination pair combinations and general routing. We will also assume that the number of neighbours of all the nodes are same, i.e., $n_i = n, \ \forall i$. Also, we will be assuming that $K_{i,s,d} \equiv 1$. Note that assuming a symmetric network need not imply that the number of nodes is infinite. *We mention that the restriction to symmetric case is only to simplify the presentation and all the following development will work for a general network as well.*

We give some necessary and some sufficient conditions for stability of the forwarding queues. These stability conditions can be grouped into two category: (i) stability conditions specific to a particular routing, and (ii) stability conditions independent of the routing.

Clearly, the stability conditions which account for routing will give tighter conditions. However, obtaining stability conditions that do not depend on the routing is in itself significant simplification in tuning the network parameters. For example, suppose that we are deploying a grid (or, mesh) network for which $n_i = 4$. In this case, if we can find a pair of values P an P_f such that *all the forwarding queues are guaranteed to be stable*, then one can decouple the problem of finding an optimal route and that of stability. We will use this decoupling later in the paper.

Let $r \overset{\Delta}{=} (1 - P)^n$. Note that $P_{i,s,d} = r$. Also, for a given routing, let $d(i, s, d)$ be the number of elements in the set $Ri, s, d \backslash i$.

4.1 Stability Conditions

Proposition 2. *1- A* necessary *condition for stability of F_i for a given routing is that*

$$PP_f \geq \sum_{s,d:i\in R_{s,d}} (1-P_f)P_{s,d}Pr(1-r)^{d(i,s,d)}.$$

2- A sufficient *condition for stability of F_i,* irrespective of routing *is that*

$$PP_f \geq (1-P)^n.$$

Proof: 1- For a given routing, the input rate into the forwarding queue F_i is

$$\sum_{s,d:i\in R_{s,d}} y_s PrP_{s,d}(1-r)^{d(i,s,d)}.$$

Now, $y_s = 1-\pi_s P_f \geq 1-P_f$. Hence, the minimum rate at which packets can arrive to F_i is

$$\sum_{s,d:i\in R_{s,d}} (1-P_f)PrP_{s,d}(1-r)^{d(i,s,d)}.$$

The maximum rate at which F_i can be served is clearly PP_f. The proof is complete for 1.

2- The maximum arrival rate of packets into the queue F_i is $(1-P)^n = r$, because in any slot F_i can receive packet only if the node i and $(n-1)$ of its neighbours are not transmitting. Similarly, the maximum rate at which the queue F_i is served is PP_f. For stability we need the service rate to be at least the arrival rate. The proof is complete. •

4.2 Effect of Routing

Assume a symmetric network and assume that the condition of Proposition 2 is satisfied so that all the forwarding queues are always stable, irrespective of the routing of packets.

Under the present situation where stability is guaranteed irrespective of the routing used, we can change routing to obtain better throughput for the various routes while maintaining stability of the forwarding queues.

The probability that a packet on route $R_{s,d}$ reaches its destination is $r^{d(d,s,d)}$. Here, the quantity $d(d,s,d)$ depends on the routing used. We then have the following easy result

Lemma 3. *Shortest path routing maximizes the probability of success of a packet between a source-destination pair.*

Proof: From the expression of probability of success of a packet on a route, we need minimum value of $d(d,s,d)$ to maximize the probability. •

The above result was fairly straightforward to obtain and is also intuitive. It is similarly easily shown that

Corollary 1. *If number of neighbours is not same for all the nodes then a route with shortest number of interfering nodes achieves maximum probability of success of packet.*

Even though we are able to ensure that the forwarding queues are stable independent of the routing used, it is clear that maximizing the probability of success of a packet on any route does not necessarily maximize the *throughput* on that route. This is because the throughput on a route $R_{s,d}$ is $y_s PrP_{s,d} r^{d(d,s,d)}$, so that it is possible that the probability of success on a route increases but the forwarding queue of the source itself is loaded so much that the throughput that the source decreases.

However, we know that the minimum rate at which queue Q_s is served is $P_s(1-f_s) = P(1-P_f)$, independent of the load on queue F_s. Hence, by maximizing the probability of success for each source-destination pair by using shortest-path routing maximizes the minimum guaranteed throughput for the source-destination pair. This in itself is important consequence of Lemma 3.

Remark. The results of this section deal with the effect of routing on the minimum guaranteed throughput. We assumed that the system is always stable, independent of the routing used (we also gave a sufficient condition for this to happen). However, we have not answered the question of maximizing the throughput itself. This is a hard problem in general as can be seen by the complex dependence of y_s on the routing. Moreover, assuming a shortest path routing does not always uniquely determine the routing in a network. This is because in a network there may be many paths between a given source-destination pair which qualify to be shortest path. A simple example is a Grid network. In our ongoing research work we are looking at the problem where we restrict ourselves to the space of shortest path routing and then aim at maximizing the throughput obtained by the routes. This amounts to maximizing y_s for each value of s. This also amounts to minimizing the value of π_s for each s. Clearly, this need not always be possible since two vectors need not always be component-wise comparable. Hence, we are looking at the problem of maximizing an overall utility function

$$\max_{\text{Shortest Path Routing}} \sum_s \frac{(y_s r^{d(d_s,s,d_s)})^{1-\alpha}}{1-\alpha},$$

where we assume that a source s can have at most one destination, referred to as d_s. Above optimization problem is motivated by the concept of fairness in communication networks. When $\alpha \to \infty$, the above optimization problem aims at maximizing the minimum throughput obtained in the network. This also amounts, roughly, to minimizing the maximum value of π_s, so that all the forwarding queues in the network are uniformly well behaved.

4.3 Numerical Results: An Asymmetric Network

In this section, we study the observations made in the Section 4 by means of a simple asymmetric example network. In this example, we show that the results and observations made in Section 4 are also valid for a general network.

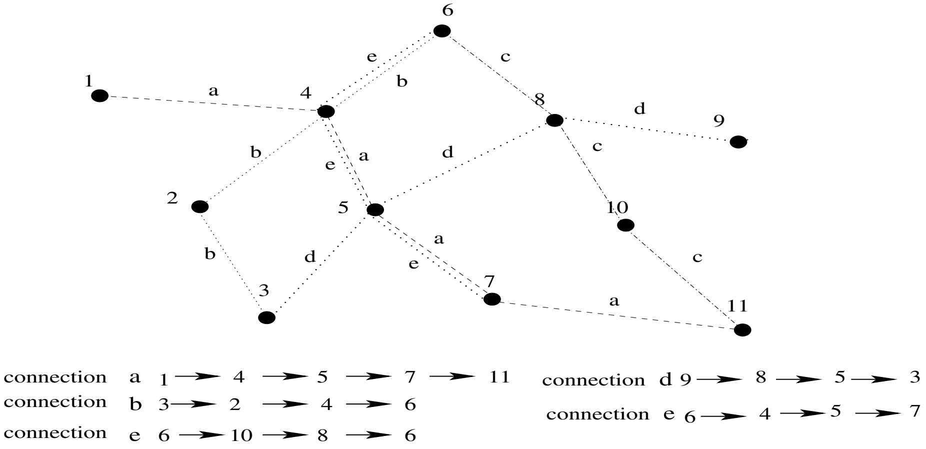

6
e
c
1
a
4
b
8
d
9
b
e
a
d
c
2
5
10
a
b
d
e
7
c
3
a
11
connection a 1 → 4 → 5 → 7 → 11
connection b 3 → 2 → 4 → 6
connection e 6 → 10 → 8 → 6
connection d 9 → 8 → 5 → 3
connection e 6 → 4 → 5 → 7

Fig. 1. The Asymmetric Network considered for studying effect of routing

Consider the asymmetric wireless ad-hoc network consisting of 11 nodes as depicted in Figure 1. We assume that there are only five end-to-end connections defined as follows : $R_{1,11} = \{4, 5, 7\}$, $R_{3,6} = \{2, 4\}$, $R_{11,6} = \{10, 8\}$, $R_{9,3} = \{8, 5\}$ and $R_{6,7} = \{4, 5\}$ where $R_{s,d}$ is the set of intermediate node used by a connection from source s to destination d.

The routing used in this example is based on hop-length in which each source selects a route with minimum number hop. To ensure the stability of the example network under consideration, we fix the channel access probabilities of nodes 4, 5, 8 and 10 to 0.3. The channel access probability of the other nodes are equal. In figures 2, we plot the throughput on various routes and the quantities y_i, $i \in \{2, 4, 5, 7, 8, 10\}$[2] against the channel access probability for the different values of limits on attempts (assuming $K_{i,s,d} \equiv K$ and $P_i = P$ for $i = 1, 2, 3, 6, 9, 11$). The existence of an optimal channel access rate (or, the transmission probability) is evident from the figures. Moreover, as expected, the optimal transmission probability increases with K. By comparing the throughput and the quantities y_i for different values of $K = 1, 4$. The existence of an optimal choice of the channel access probability is evident from the figure. The figure 2 shows 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 limit on the number of attempts (K)

Now, using the same example network, we study the effect of routing on stability (as studied in section 3.2). In this example, we observe that the nodes 1, 3, 6, 9 and 11 don't forward packets from any of the connections, hence the forwarding queues of intermediate nodes that forward packets originating from these sources are less stable and become unstable when the limit on number of attempts K becomes large. To validate this observation, we added in the network

[2] Since the nodes 1, 3, 6, 9 and 11 don't forward packets of any connections, i.e., $y = 1$, we don't need to plot the quantity y for these nodes.

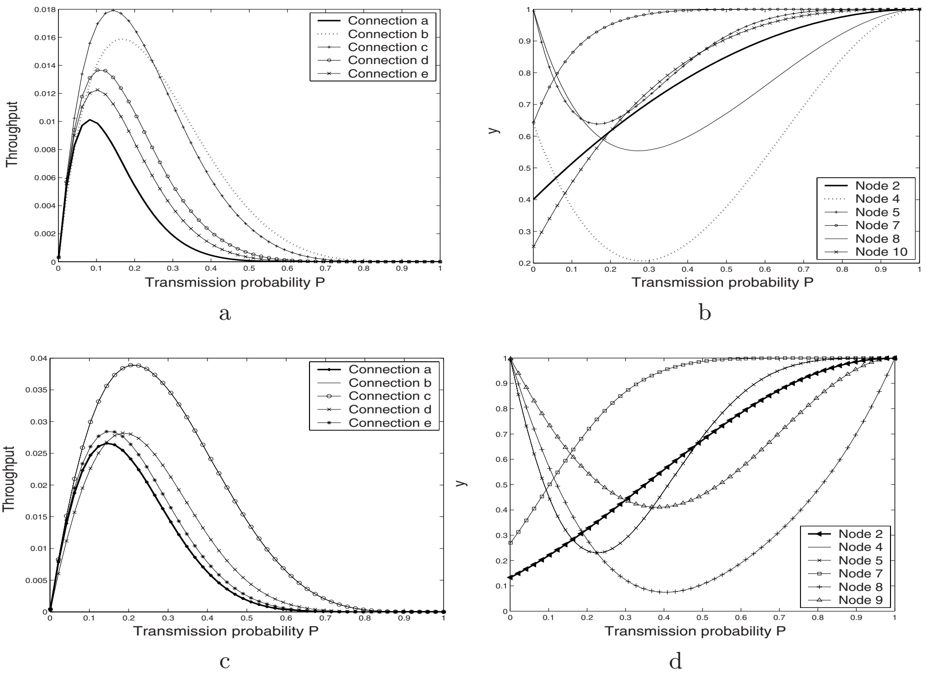


Fig. 2. (a) and (b) (resp. (c) and (d)) show he throughput of all sources and region of stability as function of the transmission probability P for $K = 1$ (resp. $K = 4$)

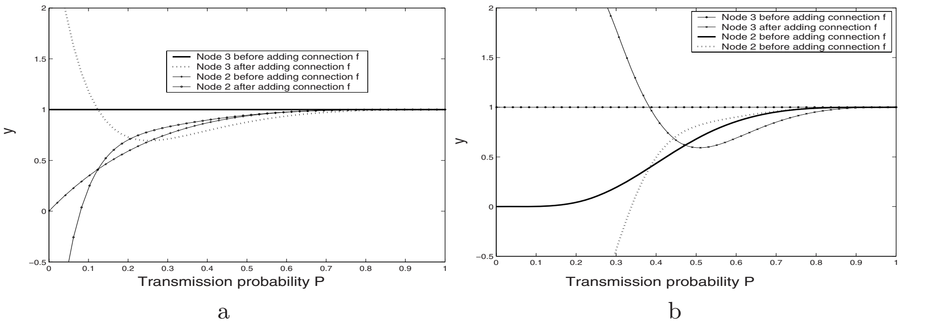
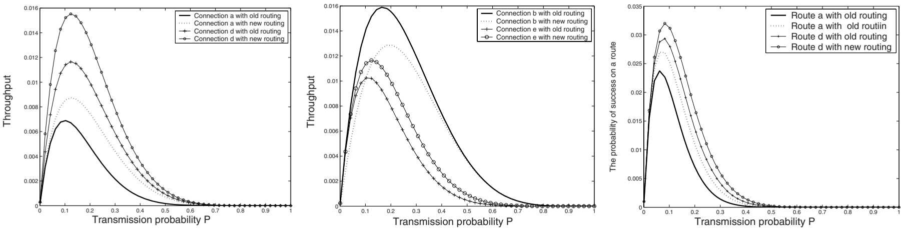


Fig. 3. (a) and (b) show the region of stability of node 2 and 3 as function of the transmission probability P for $K = 1, 4$

(Figure 1) a connection f between node 5 and node 2 such that $R_{5,2} = \{3\}$. This implies that node 3 forwards packets originating at source node 5.

In figure 3, we compare the region of stability of node 2 and node 3 before and after adding the connection f. Clearly, the forwarding queue at node 2 becomes more stable when the node 3 starts forwarding packets of connection f. This confirms our observation of Section 3.2 that the stability of the network when all source forward data is more as compared to the case when some nodes are not source of packets. Thus nodes in a random access network have a natural incentive to forward data.

Now, we use the shortest path routing (based on the number of interferers on a path as defined in subsection 4) under the present situation where the stability of all the forwarding queues in the network is guaranteed. The routes for all connections under this shortest-path routing are $R_{1,11} = \{2, 3, 7\}$, $R_{9,3} = \{10, 7\}$ and $R_{6,7} = \{8, 10\}$.



a b c

Fig. 4. (a) and (b) show the throughput of connections a, c, b and e as function of the transmission probability P for $K = 1$ and (c) shows the probability of success on a route a and d as function of the transmission probability P for $K = 1$

In figures 4 ((a) and (b)), we compare the throughput of all connections under the old and new routings. We observe that the throughput of all connections (except that of connection b), is better with new routing than those obtained under the old routing. The reason of decreasing the throughput of connection b is the change in quantity y_3. In old routing, $y_3 = 1$ (node 3 with old routing, does not forward packets of any connections). With the new routing, node 3 forwards the packets of connection a. However, the value of y_3 decreases with new routing, explaining the decrease of throughput of connection b (because now the source node of connection b, i.e., node 3 gives some of its resources to forwarding of packets on route a). In conclusion, the question of maximizing the throughput *uniformly for all nodes* is a hard problem. The complexity of this problem comes from the dependence of throughput and the quantity y. In figure 4 (c), we plot the probability of success of a packet on all connections versus the transmission probability P. We observe that, as predicted already in Section 4, the new routing improves the probability of success of *all* connections.

Remark 1. Studying an asymmetric network numerically requires one to consider all possible combinations of the network parameters. Since the degree of freedom (the parameters to choose) are usually very large in asymmetric networks, such a numerical study is not carried out generally.

In the full version of our paper [2], we also study some special cases as a symetric networks. In a symmetric network we have $n_j = n$ for all nodes; some examples are a grid network, a circular network or a linear network. Moreover, for the symmetric networks, we can simplify the expressions in the detailed balance equation (Proposition 1) while getting important insights into the working of the network.

5 Conclusion

Considering a simple random access wireless network we obtained important insights into various tradeoffs that can be achieved by varying certain network parameters.

Some of the important results are that

1. As long as the intermediate queues in the network are stable, the end-to-end throughput of a connection does not depend on the load on the intermediate nodes.
2. Routing can be crucial in determining the stability properties of the network nodes. We showed that if the weight of a link originating from a node is set to the number of neighbors of this node, then shortest path routing maximizes the minimum probability of end-to-end packet delivery.
3. The results of this paper extended in a straighforward manner to systems of weighted fair queues with coupled servers.

References

1. V anantharam, "The stability region of the finite-user slotted Alloha protocol, III Trans. Inform. Theory, vol. 37, no. 3, pp. 535-540, May 1991
2. A. Kherani, R.El Azouzi et E. Altman "Stability-Throughput Tradeoff and Routing in Multi-Hop Wireless Ad-Hoc Networks `http://www.lia.univ-avignon.fr/php/publications2.php?page=5&selection=auteur&tableau2=elazouzi`
3. B. Radunovic, J. Y. Le Boudec, "Joint Scheduling, Power Control and Routing in Symmetric, One-dimensional, Multi-hop Wireless Networks," WiOpt03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Sophia-Antipolis, France, March 2003
4. T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum and L. Viennot, "Optimized Link State Routing Protocol" IEEE INMIC Pakistan 2001.
5. M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Adhoc Wireless Networks", IEEE/ACM Transactions on Networking, vol. 10, no. 4, August, 2002, pp. 477-486.
6. P. Gupta and P. R. Kumar, "The capacity of wireless networks," III Trans. Inform. Theory, vol. 46, no. 2, pp. 388-404, March, 2000
7. X. Huang, B. Bensaou. "On Max-min fairness and scheduling in wireless Ad-Hoc networks: Analytical framework and implementation," In proceeding MobiHoc'01, Long Beach, California, October 2001
8. S. Murthy and J. J. Garcia-Luna-Aceves "An Efficient Routing Protocol for Wireless Networks", ACM/Baltzer Journal on Mobile Networks and Applications, Special Issue on Routing in Mobile Communication Networks, vol. 1, 1996.
9. Sorav Bansal, Rajeev Shorey and Arzad Kherani, "Performance of TCP and UDP Protocols in Multi-Hop Multi-Rate Wireless Networks", in IEEE WCNC, Atlanta, USA, March 2004.

10. W. Szpankowski, "Stability condition for some multiqueue distributed systems: buffered random access systems," Adv. Appl. Probab., vol. 26, pp. 498-515, 1994.
11. L. Tassiulas and A. Ephremides, "Stability properties of constrained queuing systems and scheduling for maximum throughput in multihop radio network", IEEE Trans. Automat. COntr. vol. 37, no 12, pp. 1936-1949, December 1992.
12. L. Tassiulas, "Linear complexity algorithm for maximum throughput in radio networks and input queued switches," in IEEE Infocom 98, pp. 533-539, 1998.
13. L. Tassiulas and S. Sarkar. "Max-Min fair scheduling in wireless networks", In proceeding of Infocom'02, 2002
14. B. S. Tsybakov and V. L. Bakirov, "Packet transmission in radio networks", Probl Infom. Transmission, vol. 21, no. 1, pp. 60-76, Jan.-Mar. 1985

EASR: An Energy Aware Source Routing with Disjoint Multipath Selection for Energy-Efficient Multihop Wireless Ad Hoc Networks[*]

Do-Youn Hwang, Eui-Hyeok Kwon, and Jae-Sung Lim

Graduate School of Information and Communication, Ajou University, South Korea
{soyosoyo, k31001, jaslim}@ajou.ac.kr

Abstract. Wireless ad hoc networks usually consist of mobile battery operated computing devices that communicate over the wireless medium. These devices need to be energy conserving so that the battery life is maximized. The energy for transmission of a packet in the wireless channel remains quite significant and may turn out to be the highest energy consuming component of the device. So, an energy-efficient communication protocol can minimize maintenance and maximize system performance. We propose an Energy Aware Source Routing (EASR) which can be efficient from network long-term connectivity point of view. In this algorithm, multiple routing paths are selected. However, only one path will be used for data transmission at a certain time among multiple paths and each path has probability to be selected. In EASR, the routing paths will be discovered without overlapped. In addition, each path hardly overhears other data transmission. We define an *overhearing ratio* in order to reduce the overhearing energy waste among each selected path. And we show how establish energy efficient multiple paths by making use of *overhearing ratio*. Our simulation results show that our proposed scheme can achieve magnitude improvement of network lifetime and reasonable packet latency time.

Keywords: EASR, energy-efficient, overhearing ratio, multipath.

1 Introduction

In wireless ad hoc networks, the battery of the network devices may not be replenished. So, energy-efficient communication protocol is the most important key to prolong network lifetime. Almost all of conventional routing protocols in wireless ad hoc networks find only a single optimal path and use it for every communication. It reduces the delay of data communication. However, any single path is apt to be disconnected by energy depletion. If a communication route breaks off, then we have to discover another route to maintain data transmission from source to destination. This phenomenon brings about more number of trials finding another route. As a result, we hardly prolong the networks connectivity.

The Energy Aware Routing (EAR) is one of the best known protocol selecting sub-optimal paths according to the probability calculated by energy metric [1]. Energy

[*] This work was supported by the Korea Research Foundation Grant (KRF-2004-013-D00028).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 41 – 50, 2006.

depletion can be prevented. However, it has another problem that the packet latency time can be increased because in EAR network nodes know sub-optimal single hop path to distribute traffic load. Therefore, we use multiple paths those are source-to-destination paths in this paper. This is called Energy Aware Source Routing (EASR) protocol.

We tried to find multiple paths using the route discovery scheme of DSR [2], but almost all of the found paths are overlapped because all of the duplicated RREQs are dropped by intermediate node. So, we need another way to find maximally disjoint paths. We can use the route discovery procedure of Split Multipath Routing (SMR) which introduces the different way of route discovery instead of dropping every duplicate RREQs [3]. The intermediate nodes forward the duplicate packets which traversed through a different incoming link. By this way we can establish disjoint multipath from source to destination. However, although they are not overlapped, the overhearing effect among paths can be occurred. It causes energy waste of each selected path. Figure 1 shows the overhearing effect among nodes on each selected path. In the figure, {s, j, i, h, g, f, d} is a set of nodes that belong to the path 1 between nodes *s* and *d* and data is transmitted on it. Assume that other paths 2 and 3 are set up by {s, l, m, n, o, p, q, d} and {s, a, b, c, k, e, d}. Some nodes {l, m, n, p, q, c, k, e} can overhear from the path 1 communication, then the energy of the node is wasted due to overhearing. The multiple paths communication has still the unnecessary energy waste problem.

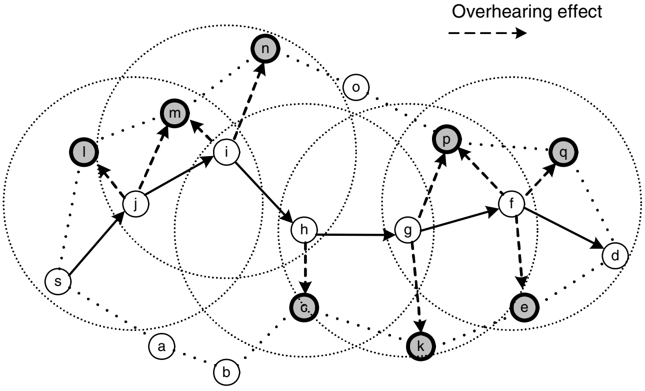


Fig. 1. Overhearing effect between two paths

Consequently, we present an energy aware source routing with a disjoint multipath selection scheme. This protocol means that data communication would use different paths at different time, and not only those paths are not overlapped but also the overhearing effect can’t be occurred among the paths. We design a simple route discovery procedure to solve overhearing problem. And we define an *overhearing ratio* to find out the level of energy waste from overhearing effect between the paths. The source can select multiple routes by measuring *overhearing ratio* of each path. If the *overhearing ratio* is too high, the path will not be selected by the source. It can increase the network lifetime. In addition, providing multiple paths is profitable in wireless ad hoc network where routes are failed frequently.

The remainder of this paper is organized as follows. Section 2 describes the related works. And we introduce our EASR protocol in Section 3 in detail. Simulation results are presented in Section 4 and concluding remarks are made in Section 5.

2 Related Works

Finding an optimal single path is a normal approach of routing protocol for wireless ad hoc networks but it needs to be changed to prolong network connectivity.

Dynamic Source Routing (DSR) is an on-demand routing protocol for wireless ad hoc networks [2]. If any nodes want to send data packet, they flood a route request (RREQ) into the network. Any node that has a path to the destination can reply with the route reply (RREP) to the source. RREP packet contains the entire path, so the data packet can be sent properly. Almost all of the ad hoc routing protocols always use a single optimal path. Even though they use a single energy efficient path, which can deplete node energy locally and so the network is easy to be disconnected. Consequently, the source needs to flood RREQ into the network to discover another path again.

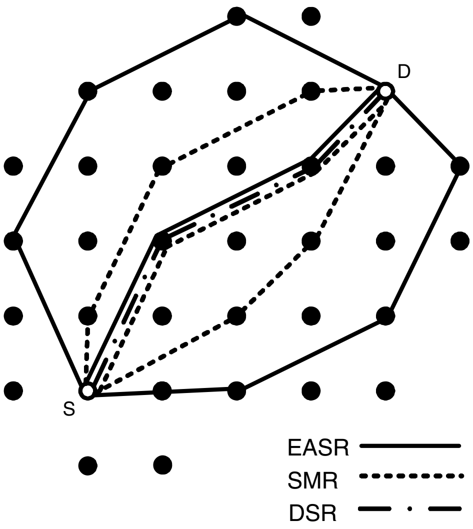


Fig. 2. Results of the route discovery of the various protocols

Split Multipath Routing (SMR) was proposed for wireless ad hoc networks [3]. The route discovery scheme of SMR is the best way to find maximally disjoint paths. It uses the source routing protocol approach where the information of the nodes that consist of the route is included in the RREQ packet. In SMR, instead of dropping every duplicate RREQ, the intermediate nodes forward the duplicate packets, if they are traversed through a different incoming link. And the hop count of new arriving RREQ is not larger than that of the first received RREQ. However, although they are not overlapped, we have observed in our experiments that the overhearing effect among paths is occurred. Figure 2 shows the results of the route discovery procedure of the various protocols.

3 Proposed EASR Protocol

3.1 Route Discovery

The Energy Aware Source Routing (EASR) is an on-demand routing protocol that builds multiple paths using request/reply cycles. In our algorithm, the source can select multiple disjoint paths so that the overhearing effect among paths would not be occurred. The route discovery procedure of EASR is similar to that of SMR. Disjoint multiple paths can be maintained from source to destination. However, SMR selects the paths which are just disjoint only, while EASR selects the paths which can avoid overhearing effect among them using the neighbor information that is gathered from the route reply (RREP). Whenever the source transmits data to the destination, the source chooses one of the good paths according to probabilistic fashion to prevent energy depletion.

Route Request. Our scheme has to discover maximally disjoint paths in order to avoid unnecessary energy consumption of nodes on the paths due to overhearing effect among paths. So we use the same procedure of the route request of SMR. In SMR, instead of dropping every duplicate RREQs, the intermediate nodes forward the duplicate packets that traversed through a different incoming link. And the new arriving RREQ has to also satisfy the condition that the hop count of the duplicated packets is not larger than that of the received RREQ. It is worth to note that the route request of SMR is one of the best ways to find maximally disjoint paths in spite of the increase of the number of RREQ.

Route Reply. In our protocol, we assume that all of the nodes collect neighbor node IDs. In the on-demand routing protocol, the node IDs on the entire path are recorded in the RREP, and hence the intermediate nodes can forward the RREP using this information. In addition, EASR requires additional information for establishing energy efficient multiple paths. The protocol has two phases:

a) The residual energy of all intermediate nodes is recorded in the RREP.
b) The neighbor list of each node is recorded in the RREP.

The neighbor list of each node is necessary to select good paths in the following route selection procedure. And the sum of residual energy of each path is gathered at the source node. A path selection probability of each path for the data transmission procedure will be calculated using the sum of residual energy of each path.

Route Selection. The main goal of our scheme is selecting energy efficient multiple paths from the point of view of energy-efficiency. We want to construct at least two paths that are not overlapped. Furthermore, each path has to keep a proper distance from the other paths to avoid overhearing effect among paths. Thus, we need a different procedure from the conventional on-demand routing protocol. First, the source node records a path of the first received RREP at the routing cache. Second, after the source receives other RREPs, the source decides whether the path information of RREP will be selected or not. If the path of RREP is overlapped or has experience in the overhearing effect, the RREP will be dropped. Except only the first arriving RREP, all of the new arriving RREPs should be considered whether they are selected or not. The following conditions make a new RREP is dropped.

a) The path of a new arriving RREP is overlapped with other paths that are selected already.

b) The neighbor lists of a new arriving RREP are overlapped with other paths that are selected.

The first condition is absolutely necessary, so all RREPs included in this condition must be dropped in the source node. However, the second condition is flexible. We can adjust a threshold of *overhearing ratio* defined as

$$R_k = \frac{\sum_{x \in S_{new}} |N_x \cap S_k|}{|S_k|} \tag{1}$$

where R_k is the *overhearing ratio* of the *k*th path which is already on a routing cache of the source node, in other words, the source node are already using the path for data transmission. And N_x is a neighbor list of the node *x*, S_k is the set of intermediate nodes which are included in the *k*th path. Additionally, S_{new} is the set of intermediate nodes of new arriving RREP, and $|S_k|$ is the number of elements of set S_k.

Figure 3 shows the new arriving RREP can be dropped because of the overhearing effect. In the figure, S_1 is {a, b, c}, S_2 is {e, f, g} and S_{new} is {l, k, j, i, h}. Next, N_h, N_i,

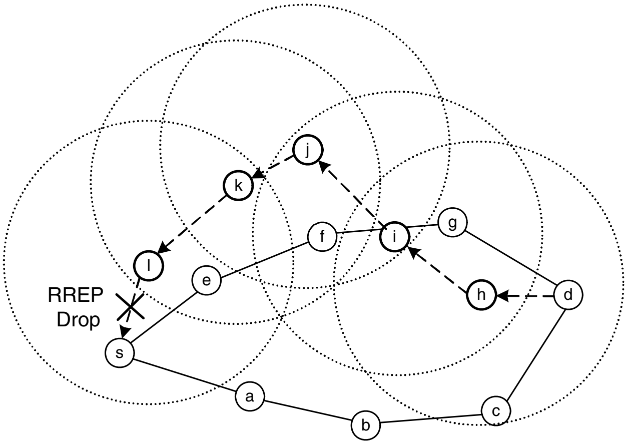


Fig. 3. An example of the EASR route selection

N_j, N_k, N_l are neighbor list set of S_{new}, for example N_h is {i, g, c}. By applying (1) to this case, we can acquire the value R_1 is 0.33 and R_2 is 2.33. Then, the source node needs to decide whether to select the path of new arriving RREP or not. In the case of the figure 3, if we set the threshold of *overhearing ratio* to 0.4, the new arriving RREP should be dropped. Since, the value R_2 is larger than threshold 0.4. That is to say, accepting a new path requires that all of the *overhearing ratios* of each path are lower than the threshold. Because *overhearing ratio* indicates the extent of the energy waste due to overhearing effect of selected paths.

3.2 Data Transmission

The source node sends data packets to the destination with the probability of the each selected path. This means that none of the paths is used all the time to prevent energy depletion. Thus we use a similar way with EAR. But EASR is a source routing scheme, thus we can consider the number of hops from source to destination because of delay perspective. Consequently the end-to-end packet transmission delay is lower than that of EAR. Each path is assigned by the probability of path chosen according to a simple metric given by

$$P_n = E_n \Big/ \sum_{k=1}^{S} E_k \tag{2}$$

where P_n is the probability of nth path, E_n is energy metric of the nth path, and S is the number of whole selected routes. This simple metric can contribute low latency with high energy efficiency. As we mentioned before, the residual energy of all intermediate nodes is recorded in the RREP, so the source node can compute probability of each path. This energy metric E_k computed as

$$E_k = (R_k)^{\alpha} \cdot (1/n_k)^{\beta} \tag{3}$$

where R_k is the average residual energy of all nodes on the kth path, n_k is the number of hops on the kth path. The weighting factors $\alpha$and βcan be chosen to find the energy efficient path or the path with low latency.

The energy metric is a very important component of the proposed protocol. Depending on the metric, the characteristics of our scheme can be changed substantially. We use a simple metric that can reduce delay and energy waste.

4 Simulations and Results

The simulations were carried out in NS-2.28 to evaluate the proposed scheme in terms of network lifetime. Simulation model is defined as a network of 50 mobile hosts placed randomly within a 1000 meter × 1000 meter area, but the position of the sources and destinations is fixed in proper place in the simulation area. 5 sources and 5 destinations are selected randomly. Each source starts the data transmission at random time and it will stop after 100 seconds. However, the sources and destinations can not be neighbor of each other because in that case no routing protocol is needed. Table 1 shows the simulation parameters in detail.

Table 1. Parameters of NS-2.28 simulators

Parameters	Value
Packet size	30 bytes
Number of nodes	50
TxPower	30 mW
RxPower	20 mW
Initial node energy	30 Joules
(α, β)	(1, 0.1)
Overhearing ratio threshold	0.4/0.7/1.5
Packet arrival rate (packet/s)	17 ~ 167
MAC protocol	IEEE 802.11b
Simulation time	500 seconds

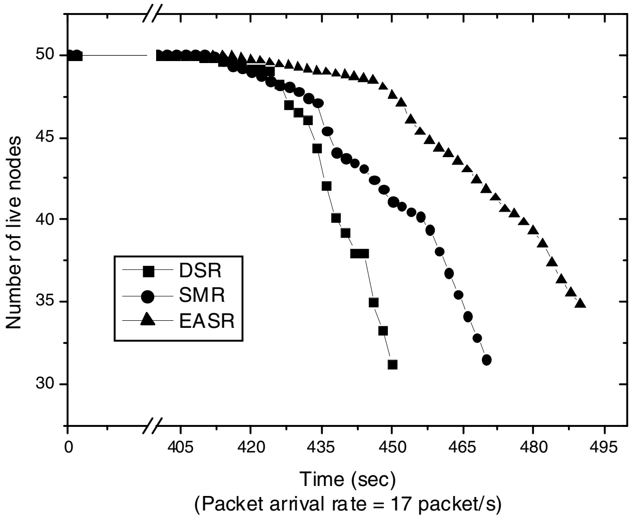


Fig. 4. Number of live nodes in networks

Figure 4 illustrates the number of live nodes with each protocol. We can see that EASR can prolong the network connectivity longer than comparable schemes such as DSR and SMR. In this simulation, we choose 0.4 for a threshold of *overhearing ratio*. It means that the overhearing effect is allowed only 40% of the entire path. Almost all of conventional ad hoc routing protocols such as DSR always use a single route. Even though they choose an energy efficient path, they can deplete the nodes energy locally. The path is susceptible to be disconnected. Therefore the source node of DSR needs to flood RREQ into the network to find another path again. It is observed that the number of live nodes is decreased dramatically when the route discovery procedure of DSR is used in networks. And, at 450 seconds, the source can't find any route to destination anymore. Unlike DSR, EASR discovers multiple paths and uses them for data transmission to prevent energy depletion. In addition the source node sends data to destination with the probability of each selected path. So we can see better performance when we compare EASR with SMR and DSR. In short, the performance of EASR is the best because the traffic load is distributed to multiple paths and each path can avoid overhearing energy waste.

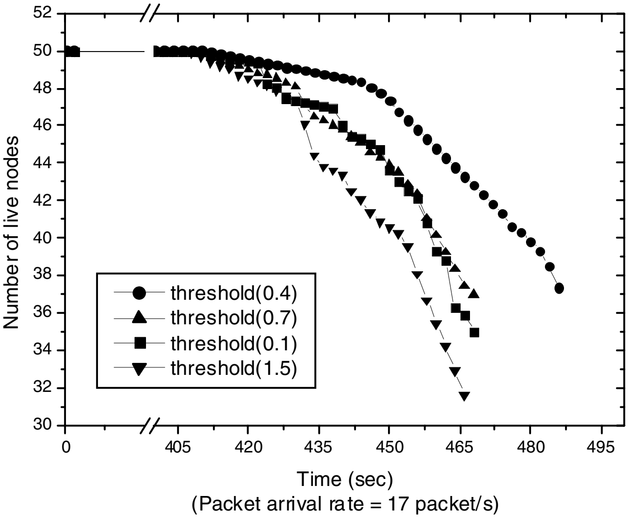


Fig. 5. Number of live nodes according to threshold of overhearing ratio of EASR

Obviously, if we apply different threshold of *overhearing ratio*, the performance of EASR would be changed. Figure 5 shows the number of live nodes of EASR according to the threshold of *overhearing ratio*. The large threshold value means that most of the paths can be selected for data communication even though wasted energy due to overhearing effect is serious. Otherwise, the small threshold value means that the overhearing effect is not allowed strictly. One of the most important things is adjusting threshold level. The small threshold can achieve good performance, but if the threshold is extremely small, the source can establish only one path in most of cases. Therefore it operates like DSR. As we saw the figure 5 the performance of threshold (0.1) is worse than that of threshold (0.4). In addition, if the threshold value is overly big as 1.5, then EASR operates like SMR.

Figure 6 illustrates the end-to-end packet transmission time of each protocol. And we choose 0.4 for a threshold of *overhearing ratio*. Because DSR discovers a single shortest path, the performance is better than other schemes when the packet arrival rate is lower than about 80pps. However, according to increase of data rate, queuing delay is raised. The queuing delay makes the performance of each protocol worse. Especially, the end-to-end packet transmission delay of DSR is increased dramatically according to increase of data rate because DSR uses only a single shortest path for every data transmission. On the other hand, the data traffic is split into multiple routes in the case of SMR and EASR. Therefore the end-to-end packet transmission delay of SMR and EASR is increased slowly according to increase of data rate.

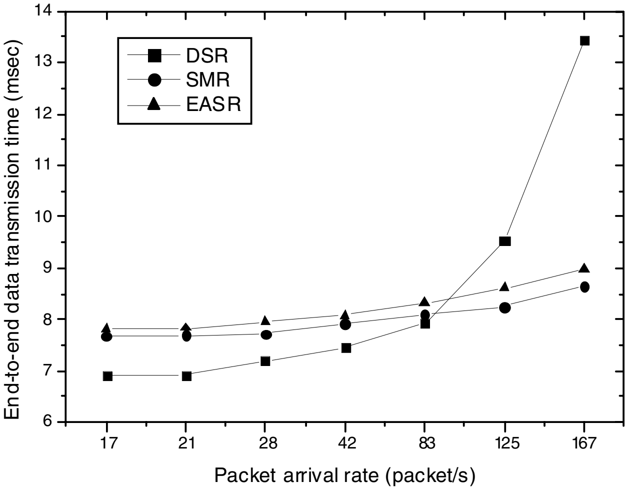


Fig. 6. End-to-end packet transmission time according to packet arrival rate

5 Conclusions

In this paper, we presented the Energy Aware Source Routing (EASR) protocol for energy-efficient wireless ad hoc networks and analyzed its performance through simulations. We introduced the *overhearing ratio* as a proper factor to control overhearing effect of each path. Additionally data propagation of EASR is performed like EAR in order to saving battery energy of ad hoc device. However, EASR is on-demand source routing protocol, therefore the delay aspect is enhanced compared with EAR obviously. Besides, maintaining multiple paths is useful in wireless networks because the source can simply use other available route without performing the route recovery process.

References

1. Rahul C. Shah and Jan M. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks”, IEEE Wireless Communication and Networking Conference, 2002
2. D. B. Johnson and D. A. Marltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, 1996, pp. 153-181
3. S. J. Lee and M. Gerla, “Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks”, ICC 2001.
4. C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A scalable and robust communication paradigm for sensor networks”, IEEE/ACM Mobicom, 2000, pp. 56-67.
5. R. Ogier, V. Rutenburg, and N. Shacham, “Distributed Algorithms for Computing Shortest Pairs of Disjoint Paths,” IEEE Transactions on Information Theory, vol. 39, no. 2, Mar. 1993, pp. 443-455.
6. S. Singh, M. Woo, and C. S. Raghavendra, “Power aware routing in mobile ad hoc networks”, IEEE/ACM Mobicom, Oct. 1998, pp. 181-190.

7. J. Chang and L. Tassiulas, “Energy conserving routing in wireless ad hoc networks”, IEEE Infocom, 2000, pp. 22-31.
8. Feeney, L., Nilsson, M., “Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment,” IEEE INFOCOM 2001
9. J. Chou, D. Pelrovis, and K. Ramchandran. “A distributed and adaptive signal processing approach to reducing energy consumption in Sensor networks:’ in Proc. IEEE INFOCOM 2003, April 2003, San Francisco, CA.
10. S. Doshi, S. Bhandare, and T.X. Brown. “An on-demand minimum energy routing protocol for a wireless ad hoc network: ACM Mobile Computing and Communications Review. vol. 6. no. 3, July 2002.
11. S. Rhee, D. Seetharam and S. Liu. “Techniques for Minimizing Power Consumption in Low Data-Rate Wireless Sensor Networks” Wireless Communications and Networking Conference 2004, IEEE Vol. 3, Mar. 2004, pp. 1727 – 1731.
12. X. Li, “Energy efficient wireless sensor networks with transmission diversity,” *Electron. Lett.*, vol. 39, no. 24, pp. 1753–1755, Nov. 2003.

On Improving the Accuracy of OSPF Traffic Engineering*

Gábor Rétvári, József J. Bíró, and Tibor Cinkler

High Speed Networks Laboratory,
Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics,
H-1117, Magyar Tudósok körútja 2., Budapest, Hungary
{retvari, biro, cinkler}@tmit.bme.hu

Abstract. The conventional forwarding rule used by IP networks is to always choose the path with the shortest length – in terms of administrative link weights assigned to the links – to forward traffic. Lately, it has been proposed to use shortest-path-first routing to implement Traffic Engineering in IP networks, promising with a big boost in the profitability of the legacy network infrastructure. The idea is to set the link weights so that the shortest paths, and the traffic thereof, follow the paths designated by the operator. Unfortunately, traditional methods to calculate the link weights usually produce a bunch of superfluous shortest paths, often leading to congestion along the unconsidered paths. In this paper, we introduce and develop novel methods to increase the accuracy of this process and, by means of extensive simulations, we show that our proposed solution produces remarkably high quality link weights.

Keywords: OSPF, traffic engineering, linear programming, shortest paths.

1 Introduction

OSPF Traffic Engineering (OSPF TE) exploits the potential of the Internet network infrastructure to implement economic and efficient traffic management right at the IP level. IP routers traditionally forward traffic along the shortest path(s) towards the destination, where the path length is computed in terms of an administrative weight associated with network links, and load-balancing is achieved by the optional Equal-Cost-MultiPath (ECMP) technique, that allows the traffic to be split roughly evenly amongst equal cost shortest paths. OSPF TE basically means the careful manipulation of OSPF link weights aiming towards balanced traffic distribution and reduced congestion [1], [2], [3].

The process model of OSPF TE is as follows. A dedicated TE network component participates in the signaling of the Open Shortest Path First (OSPF, [4]) or Intermediate-System-to-Intermediate System (IS-IS) routing protocol. Based

[*] This work has been done as a part of the European sixth framework research project IP NOBEL (www.ist-nobel.org).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 51–62, 2006.

on the routing information gathered from the network it becomes possible to compute OSPF link weights, so that the resultant shortest paths manifest some sophisticated TE goal. After the link weights are distributed back to the routers, the traffic in the network will follow the paths assigned by the traffic engineer, leading to, hopefully, more optimal network utilization and better user experience. And all this happens without modifying the basic operation of OSPF/IS-IS in any regards. This is in sharp contrast to the conventional models for traffic engineering, where TE functionality is delegated to a dedicated connection-oriented infrastructure, that has to be purchased, operated and maintained separately from the IP layer.

A fundamental restriction of OSPF TE is that all traffic must follow the shortest paths in the network. Only if a path set can be represented as a set of shortest paths, that is, positive, integer-valued link weights exist over which the paths are all shortest paths, it can be used in conjunction with OSPF TE. Notably, this limitation does not turn out to be overly restrictive, because any optional path set is either shortest path representable by itself, or otherwise, by eliminating redundant loops, it can be reduced to a shortest path representable one. Moreover, the reduced path set is not only capable to satisfy the same bandwidth requirements as the original path set, but it is also strictly shorter in terms of the overall number of edges traversed. This ground-breaking result, which is due to Wang *et al.* [5], immediately catapulted OSPF TE into the focus of interests, since it suggests that the range of potential path assignment strategies compatible with OSPF routing is much wider than anyone would have expected previously.

OSPF TE is generally NP-hard [1]. Thus, it is common to subdivide the process into two, mostly independent phases. In the first phase, a set of "good" paths is assigned to each ingress-egress router pair in the network (these will be referred to as *sessions* hereafter), and then, these paths are mapped to shortest paths. In a predecessor of this paper, [6], we pointed out that it is the first phase that hides the origin of exponential complexity of OSPF TE. Thus, to select the path set it is plausible to invoke some quick heuristics, such as for example the widest-shortest-path algorithm [7], which, in some way or another, promise with improving the performance of OSPF routing. What remains to be done is to map this designated path set to shortest paths with the greatest accuracy that is achievable.

Unfortunately, it has gone mostly unnoticed that this process is highly prone to ambiguity, and the resultant set of shortest paths usually contains a plethora of additional (and, unfortunately, completely superfluous) paths besides the ones designated in the first phase. Therefore, the carefully selected and fine-tuned traffic engineered path set often deteriorates into a bunch of overlapping and interfering shortest paths. In [6] we showed that, due to the adverse interference caused by traffic routed to those additional paths, the useful throughput might fall to an arbitrary small fraction of the optimal throughput realizable by OSPF TE. Thus, it is essential to devise methods to restrict the number of paths in a shortest path representation to the fewest possible. This is precisely the main purpose of the research work presented in this paper.

In Section 2, after briefly reviewing the mathematical model, we show an illustrative example, that exhibits all the shortcomings of the traditional OSPF TE methodology. Then, in Section 3, we overview the theory related to shortest path representability and in Section 4, we introduce some new concepts with the aim to restrict the number of superfluous shortest paths to the bare minimum. We also give a polynomial time algorithm, which, according to the simulation results presented in Section 5, proves itself remarkably useful in practice. Finally, in Section 6, we draw the conclusions of our work.

Due to space limitations, in the sequel we shall confine ourselves to the most basic theory. For an in depth exposition the reader is referred to [8].

2 An Illustrative Example

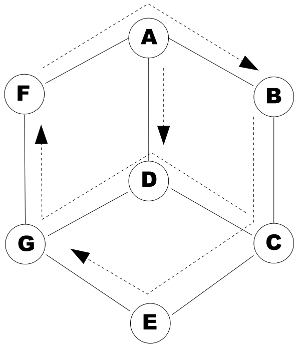
In this Section, first we introduce the relevant notation. Let the network be described by the directed graph $G(V, E)$, formed by the set of nodes V ($|V| = n$) and the set of edges E ($|E| = m$). Let N be the corresponding node-arc incidence matrix. Furthermore, suppose that we are given a set of source-destination pairs, or sessions, $(s_k, d_k) : k \in \mathcal{K}$ and, provisioned between these source-destination pairs, some set of *designated paths* $\mathcal{P}_k$. A path in $\mathcal{P}_k$ is single $s_k \to d_k$ ($s_k \in V$, $d_k \in V, k \in \mathcal{K}$) path, say, P, given by its consecutive edges: $P := \{(v_i, v_{i+1}) \in E : i = 1 \dots L_P, v_1 = s_k, v_{L_P+1} = d_k\}$, where L_P denotes the length of P. In vector-form, the *support* of P is a column m-vector p, such that the component corresponding to link (i, j) is 1 if $(i, j) \in P$ and zero otherwise. We generally assume that paths do not contain loops, and that all d_ks are distinct, which assures that IP maintains a separate entry in the routing table for each session. Let $t_k = |\mathcal{P}_k|$ denote the number of designated paths for session k.

The collection of all the designated paths is given as $\mathcal{P} = \cup_{k \in \mathcal{K}} \mathcal{P}_k$, whose support is $p = \sum_{P_k \in \mathcal{P}} p^k$ (here, p^ks are the supports of the $\mathcal{P}_k$ sets). We say that a path set $\mathcal{P}'$ is equivalent to another path set $\mathcal{P}''$, that is, $\mathcal{P}' \equiv \mathcal{P}''$ if $E(\mathcal{P}') = E(\mathcal{P}'')$, where $E(\mathcal{P}) = \{(i, j) \in E : \exists P \in \mathcal{P}, \text{ so that } (i, j) \in P\}$. Similarly, $\mathcal{P}' \subseteq \mathcal{P}''$ if $E(\mathcal{P}') \subseteq E(\mathcal{P}'')$.

Let w_{ij} be an additive, positive valued weight associated with each network link (i, j). Gather w_{ij}s into a row m-vector w. The length of a path P (of support p) over the link weights $\mathcal{W} = \{w_{ij}\}$ is defined as $W(P) = wp$. The set of shortest paths over $\mathcal{W}$ is denoted by $\mathcal{P}(\mathcal{W})$. In the remainder of this paper, link weights will be sought for in the form $w = \xi + \omega$, $\omega \geq 0$, where ξ is an strictly positive m-vector introduced with the sole purpose of separating w away from zero.

Now, we move on to introduce an illustrative example to demonstrate the main points of the paper. Suppose that we are given the network[1] depicted in Fig. 1, which we adopted from [5]. All edge capacities equal to 1. Furthermore,

[1] Note that, for the sake of simplicity, we shall use an undirected network in this example. However, the theory will be formulated for directed networks later on, which models the real case more thoroughly.



Designated paths:

(F,B): F → A → B
(A,D): A → D
(B,G): B → C → E → G
(C,F): C → D → G → F

“Plain” SPR paths:

(F,B): F → A → B
(A,D): A → D
(B,G): B → C → E → G
B → C → D → G
B → A → D → G
B → A → F → G
(C,F): C → D → G → F
C → E → G → F
C → D → A → F
C → B → A → F

mSPR paths:

(F,B): F → A → B
(A,D): A → D
(B,G): B → C → E → G
B → C → D → G
(C,F): C → D → G → F
C → E → G → F

	SPR		mSPR	
Link	weight	load	weight	load
A,B	1	1.75	2	1
A,F	1	1.75	1	1
A,D	1	1.5	2	1
B,C	1	1.75	1	1
C,D	1	1.75	1	1
C,E	1	0.5	1	1
D,G	1	0.75	1	1
F,G	1	0.75	1	1
G,E	1	0. 5	1	1

Fig. 1. Sample network topology, the set of designated paths and shortest paths in different representations and a table summarizing the weight and the emergent load of each link, assuming ECMP load-balancing

we are given 4 source-destination pairs, (namely, (F, B), (A, D), (B, G) and (C, F)) between which a set of paths, each of capacity 1, is assigned as indicated in the figure. The paths were provisioned as to assure that all links are filled to capacity. Our task is then to achieve, by the careful setting of the link weights, that all the shortest paths follow the designated paths.

Our first observation is that, by coincidence, all the designated paths are actually minimum-hop paths (so they traverse the least hops possible). Therefore, setting all the link weights uniformly to 1 assures that all the designated paths become shortest paths. This setting obviously conforms to the following conventional definition of shortest path representability [5]:

Definition 1. *A path set $\mathcal{P}$ is shortest path representable (SPR), if there exists a positive weight setting $\mathcal{W}$, such that $\mathcal{P} \subseteq \mathcal{P}(\mathcal{W})$.*

We call the attention of the reader to a subtlety in the definition. Namely, we do not require the equivalence of $\mathcal{P}$ and its shortest path representation $\mathcal{P}(\mathcal{W})$.

We only require $\mathcal{P}$ to be a subset of $\mathcal{P}(\mathcal{W})$, and quite often this is precisely the case. In our example, the "plain" shortest path representation contains significantly more paths than $\mathcal{P}$. For instance, in the case of session (B, G) not just the designated path, but also three other paths have become shortest paths. This, according to the ECMP load-balancing scheme, implies that the traffic of session (B, G) will be distributed evenly to the shortest paths, and the additional traffic directed to the superfluous paths will substantially overload some of their links. To avoid this, it is crucial to eliminate as many superfluous paths from the representation as possible. Perhaps, the most straightforward strengthening of Definition 1 would be the following:

Definition 2. *A path set $\mathcal{P}$ is perfectly shortest path representable (pSPR), if there exists a positive weight setting $\mathcal{W}$, such that $\mathcal{P} \equiv \mathcal{P}(\mathcal{W})$.*

Unfortunately, very often one can not achieve the total equivalence of the designated path set and the representation. Instead, the best one can hope for is to reduce the number of paths in the representation to the bare minimum by dropping the most paths possible. In other words, a minimal shortest path representation $\mathcal{P}_{min}$ is constituted of only those paths, which participate in *all* the shortest path representations.

Definition 3. *A weight set $\mathcal{W}_{min}$ implements a minimal shortest path representation (mSPR) of a path set $\mathcal{P}$, if for each weight set $\mathcal{W}$: $\mathcal{P} \subseteq \mathcal{P}(\mathcal{W}) \Rightarrow \mathcal{P}(\mathcal{W}_{min}) \subseteq \mathcal{P}(\mathcal{W})$. We denote $\mathcal{P}(\mathcal{W}_{min})$ as $\mathcal{P}_{min}$.*

In Fig. 1, we indicated a possible choice of weights that implements a minimal representation, and the set of shortest paths it induces. Observe that we still have superfluous shortest paths (exactly one for both (B,G) and (C,F)), but, interestingly, these paths can never be dropped from the shortest path representation. This is because, if we wanted to eliminate for example path B → C → D → G from the set of shortest paths of (B,G), we would need to increase the weight of either link (C,D) or (D,G). But in this case, the designated path C → D → G → F would cease to be a shortest path for session (C,F). Finally, it is noteworthy that using the minimal representation we could avoid to overload any of the links in the network.

In the remaining part of this paper, we shall argue that the concept of minimal representations is a remarkably useful one. First, it manifests an interesting theoretical lower bound on narrowing a shortest path representation, and, as shall be shown, this lower bound is well-defined. As a corner case, it contains perfect representations. Furthermore, a minimal representation can always be obtained in polynomial time, though, it might pose significantly more computational burden than in general.

3 A Flow-Theoretic Approach

Below, we give a brief overview of the theory of Shortest Path Representability. The fundamental result that characterizes shortest path representable paths is as follows [5], [8]:

Proposition 1. *Let $\mathcal{P} = \cup_{k \in \mathcal{K}} \mathcal{P}_k$ be a set of paths for some set of sessions $\mathcal{K}$, and let p^k be the support of $\mathcal{P}_k$ for each $k \in \mathcal{K}$. Then, $\mathcal{P}$ is representable as shortest paths, if and only if the setting $x^k = p^k$ yields an optimal feasible solution to the* primal fundamental LP of *$\mathcal{P}$, P-LP($\mathcal{P}$):*

$$\sum_{k \in \mathcal{K}} \xi p^k - \min \sum_{k \in \mathcal{K}} \xi x^k : \quad N x^k = t^k \quad \forall k \in \mathcal{K} \tag{1}$$

$$\textstyle\sum_{k \in \mathcal{K}} x^k \leq p \tag{2}$$

$$x^k \geq 0 \qquad \forall k \in \mathcal{K} \tag{3}$$

where t^k, the vector of the number of designated paths for k, is defined as:

$$(t^k)_v = \begin{cases} -t_k & \text{if } v = s_k \\ t_k & \text{if } v = d_k \\ 0 & \text{otherwise} \end{cases}$$

In this case, the optimal objective is zero. If, in contrast, the optimal objective is positive, then $\mathcal{P}$ is not SPR.

The most important observation regarding the fundamental LP is that – apart from a constant term – it basically is a minimum cost multicommodity flow problem. Constraints (1) give the flow conservation constraints with respect to t_k. The bundle constraints (2) restrict the sum of the arc-flows to remain under $\mathcal{P}$'s support, p_{ij}, at every link. In fact, p acts as some sort of link capacity. Finally, arc-flows are required to be non-negative by (3). Under the hood, P-LP($\mathcal{P}$) can be interpreted as the task to reallocate the paths in the network, such that after the reallocation the number of paths placed on a link does not exceed the number of paths using that link in $\mathcal{P}$. If this can be done such that the length of the new path set (in terms of ξ) is less than that of $\mathcal{P}$, then the path set is not SPR, because it contains loops.

The proof of the Proposition proceeds as follows. Let π^k be a row n-vector for each $k \in \mathcal{K}$, which denotes the dual variables for constraints (1). Similarly, let ω (a row m-vector) be the dual for the constraints (2). Now, for the dual variables it holds that:

$$\forall k \in \mathcal{K}, \forall (i,j) \in E : \pi^k_j - \pi^k_i \geq \xi_{ij} + \omega_{ij}, \qquad \omega_{ij} \geq 0 \ . \tag{4}$$

Hence, we could interpret π^k as distance labels, or *node potentials*, so that π^k_v defines an upper bound on the shortest distance from the source node s_k to any node v over the positive-valued weight set $\mathcal{W} = \{\xi_{ij} + \omega_{ij}\}$. The *Shortest Path Optimality Conditions* [9] require that a path P_k is shortest path from s_k to d_k, if and only if (4) holds with strict equality at all $(i,j) \in P_k$:

$$p^k_{ij} > 0 \Rightarrow \pi^k_j - \pi^k_i = \xi_{ij} + \omega_{ij} \ . \tag{5}$$

But p^k_{ij} is the dual variable corresponding to the constraints (4). Therefore, it follows from complementary slackness that (5) holds true, if and only if $[p^k]$ is optimal to P-LP($\mathcal{P}$).

Formulating the SPR problem as a multicommodity flow problem provides a wealth of options to easily solve it [10]. Furthermore, Proposition 1 gives important insights into the very nature of the problem, which we shall exploit in the foregoing discussions in our quest for improving the accuracy of shortest path representations.

4 Minimum and Perfect Shortest Path Representations

In the previous Section, we have seen that in order to obtain the link weights that represent a set of paths $\mathcal{P}$ as shortest paths, one needs to solve the dual of P-LP($\mathcal{P}$). This yields the link weights in the form: $w_{ij} = \xi_{ij} + \omega_{ij}$. However, in the vast majority of the cases there arises a large number of alternative optimal solutions, both for the primal and the dual problem, probably due to highly degenerate nature of the feasible region. The following important result characterizes the set of paths in a shortest path representation, in terms of the different alternative optimal solutions of P-LP($\mathcal{P}$).

Theorem 1. *The set of paths in the minimal representation is spanned by the alternative optimal solutions of the primal fundamental LP.*

More formally, given a set of paths $\mathcal{P}$ and some $s_k \to d_k$ path P for some session $k \in \mathcal{K}$: $P \in \mathcal{P}_{\min}$ if and only if there exists some $[x^k]$ optimal feasible solution to P-LP($\mathcal{P}$), such that $\forall (i,j) \in P : x_{ij}^k > 0$.

Proof (Sketch). Due to the *Complementary Theorem of Linear Programming* [11, p. 310, Exercise 6.39], there exists some $[x^k]$ optimal feasible solution to P-LP($\mathcal{P}$), so that $\forall (i,j) \in P : x_{ij}^k > 0$, if and only if (4) holds with strict equality at all links of P in *all* the dual solutions. Noting that, for an SPR path set, the optimal region and the feasible region of the dual coincide, this precisely means that any such P will be a shortest path over any SPR link weights. □

Regarding our example in Fig 1, the existence of additional paths in the minimal representation can be attributed to the fact that, for session (B,G) and (C,F), the subpaths of the designated paths between nodes C and G can be swapped, and both configurations supply a potential optimal feasible solution to the fundamental LP.

The significance of Theorem 1 is manifold. First, it implies that the concept of minimal representations is well-defined, because the set of alternative optimal feasible solutions of P-LP($\mathcal{P}$) is also well-defined. Therefore, the minimal representation is the intrinsic property of the path set itself, and there is *theoretically* no way to obtain a more precise shortest path representation. Another interesting corollary is that, for the single-path routing case, Theorem 1 gives a nice characterization of perfect representations:

Corollary 1. *Suppose that some path set $\mathcal{P}$ contains only one path for each distinct $(s_k, d_k) : k \in \mathcal{K}$. Now, $\mathcal{P}$ is perfectly shortest path representable, if and only if $[p^k]$ is the unique optimal feasible solution of P-LP($\mathcal{P}$).*

This result explains, why it is relatively rare for a path set to be pSPR: the support of $\mathcal{P}$ must be a unique optimizer of the corresponding fundamental LP, which, as it is usual with minimum cost multicommodity flow problems, is not a particularly frequent occurrence.

Finally, we construct an algorithm to search for the minimal representation. For this, first we introduce some more notation. Let $v^k \geq 0$ be a row m-vector of slack-variables for each session $k \in \mathcal{K}$, so that v^k is complementary to x^k in the fundamental LP. With this notation, we can write (4) as $\pi_j^k - \pi_i^k + v_{ij}^k = \xi_{ij} + \omega_{ij}$.

Corollary 2. *For some $s_k \rightarrow d_k$ path P_k: $P_k \notin \mathcal{P}_{\min}$, if and only if there exists some optimal feasible solution of the dual of P-LP($\mathcal{P}$), so that $v_{ij}^k > 0$ for some $(i,j) \in P_k$.*

Proof. From Theorem 1, $P_k \notin \mathcal{P}_{\min}$ if and only if P_k traverses at least one link, say (i,j), so that $x_{ij}^k = 0$ for each optimal feasible solution of P-LP($\mathcal{P}$). But this, from the Complementary Theorem of Linear Programming, precisely means that there exists a dual optimal solution with $v_{ij}^k > 0$, so P_k is not a shortest path according to the Shortest Path Optimality Conditions. □

In [8], we show that $v_{ij}^k > 0$ also implies that there exists an *optimal ray* d in the set of optimal feasible solutions of the dual, such that the entry in d corresponding to v_{ij}^k is strictly positive. Now, our algorithm to search for the minimal representation is based on the idea that if we elevate as many slack variables v_{ij}^k from zero as possible, while simultaneously assuring that all the designated paths remain shortest paths, we eventually obtain a minimal representation. First, we add a constraint $\sum_{k \in \mathcal{K}} v^k p^k = 0$ to the dual problem, which explicitly requires that all slack variables corresponding to the designated paths are bound to zero. Hence, solving the problem will yield yet another SPR link weight set. Second, we perturb the objective function vector by setting the cost of some v_{ij}^k to an arbitrary positive value and all other costs to zero. Finally, we set the direction of the optimization to maximization. Now, either the perturbed LP is bounded, in which case no appropriate directions exist for v_{ij}^k, or otherwise it is unbounded. Let the ray causing the unboundedness be d. Notably, d has strictly positive surplus in the position corresponding to v_{ij}^k (otherwise, the problem might not have become unbounded) and it has non-negative surplus corresponding to all other slack variables due to the non-negativity constraint imposed on the slack variables. Hence, moving along d yields a new SPR weight set, but now v_{ij}^k is separated away from zero, so all paths of k traversing (i,j) cease to be shortest paths. Repeating this step for each slack variable yields the *mSPR algorithm*:

INPUT: A designated path set $\mathcal{P}$ and initial costs $\xi > 0$.
OUTPUT: Link weight set $\mathcal{W}_{\min}$ that implements a minimal shortest path representation of $\mathcal{P}$.
THE mSPR ALGORITHM:

1. Solve the dual of P-LP($\mathcal{P}$). If the optimal objective is positive, then conclude that $\mathcal{P}$ is not SPR. Otherwise, let a feasible solution of the dual be $[\hat{\pi}^1, \ldots, \hat{\pi}^K, \hat{v}^1, \ldots, \hat{v}^K, \hat{\omega}]$.

2. For all $k \in \mathcal{K}$ and for all $(i,j) \in E \setminus E(\mathcal{P})$ with $v^k_{ij} = 0$, construct and solve the perturbed dual LP:

$$\max v^k_{ij} : \quad \textstyle\sum_{k \in \mathcal{K}} v^k p^k = 0 \tag{6}$$

$$\pi^k N + v^k = \xi + \omega \qquad \forall k \in \mathcal{K} \tag{7}$$

$$\omega \geq 0, v^k \geq 0 \quad \forall k \in \mathcal{K} \tag{8}$$

   If the perturbed problem is unbounded, then, for some optimal ray d causing the unboundedness:

$$[\hat{\pi}^1, \ldots, \hat{\pi}^K, \hat{v}^1, \ldots, \hat{v}^K, \hat{\omega}] \leftarrow [\hat{\pi}^1, \ldots, \hat{\pi}^K, \hat{v}^1, \ldots, \hat{v}^K, \hat{\omega}] + d \ .$$

3. Now, $\mathcal{W}_{\min} = \{\xi + \hat{\omega}\}$ implements a minimal shortest path representation of $\mathcal{P}$.

Interestingly, the mSPR algorithm is still a polynomial time algorithm, since the underlying solution technique remains to be linear programming. However, upgrading the definition from "plain" SPR to mSPR results in a significant complexity penalty: while computing an SPR weight set generally requires the solution of one multicommodity flow problem instance, mSPR requires $O(mK)$. Fortunately, we do not have to solve all problems from scratch: given an initial optimal feasible solution we can always start (6)-(8) from this solution, which, in the case of the two-phase simplex algorithm, eliminates all the tedious computations of the first phase. Our experiments suggest that obtaining the optimal rays is usually a matter of some few dozen simplex pivot operations. Furthermore, it is not necessary to compute a separate ray for each slack variable, because very often one ray increases multiple slack variables at once.

5 Simulation Studies

In this Section, we present the results of extensive simulation studies with the purpose of comparing different concepts of shortest path representability.

We chose to develop our SPR software toolkit in Perl, which – thanks to the unique flexibility and performance – provides an excellent platform to quickly prototype algorithms. For solving the fundamental LP we used the GNU Linear Programming Toolkit, GLPK [12]. Although GLPK does not support network programming, it is reliable, stable and, first and foremost, open source letting us to integrate the SPR toolkit very tightly into the simplex solver[2] We used the random network topology generator, BRITE [13] with the *router-level Waxman-model* ($\alpha = 0.15$, $\beta = 0.2$, $m = 3$) throughout the simulations. The source and destination node of the sessions and the capacity of the links (between 10 and 1024 units) were selected according to independent uniform distributions. Our methodology was to generate 30 random graphs with increasing number of

[2] See the `Math::GLPK` project page at `http://qosip.tmit.bme.hu/~retvari/Math-GLPK.html`.

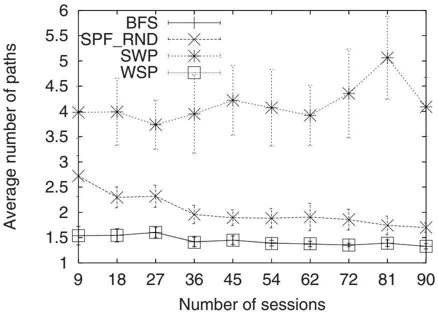


Fig. 2. Average number of paths in the SPR

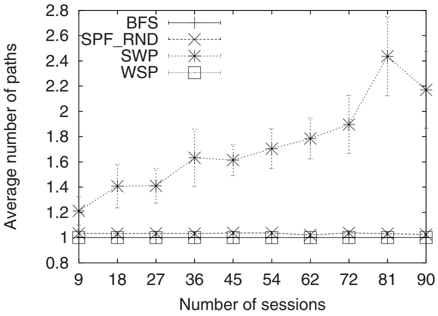


Fig. 3. Average number of paths in the *minimal SPR*

sessions and average the results (the level of significance was chosen as 95 %). Below, we present the results for networks of 45 nodes.

To select the designated paths, we used the breadth-first-search (BFS) algorithm (which manifests minimum hop-count routing), shortest path routing over random weights (SPF_RND) (which represents the case when a network operator chooses the link weights randomly) and the widest-shortest-path (WSP, [7]) and the shortest-widest-path (SWP, [14]) algorithms[3]. One may argue, why would anyone want to compute the shortest path representation of some paths, which are immediately shortest paths by themselves. For example, setting the weight of all links to 1 apparently reproduces BFS paths. The reason is that we want to observe, how many superfluous paths such a naive representation produces by comparing it with the corresponding minimal representation.

The average number of shortest paths per session in the plain SPR is depicted in Fig. 2. Note that the SPR link weights were generated by extreme point solutions of the dual fundamental LP. Our first observation is that such extreme point solutions, due to the relatively huge number of implicit zero-valued slack variables, produce low quality shortest path representations. For the WSP and the BFS paths the representation contains about one and a half times as much paths as the designated path set (which contains exactly one) almost irrespectively of the number of sessions. However, the representation of SPF_RND paths contains more than two paths in average, while this value is 4 for SWP. This suggests that a naive setting of the link weights can easily turn out to be adverse. Even if the designated paths were chosen by a SPR-compatible algorithm, such naive link weights usually only implement a superposition of a huge number random paths, and there are no appropriate mechanisms built into OSPF to select exactly the designated one from amongst them. One needs to carefully tweak the link weights to minimize the ambiguity, and this is exactly what the mSPR algorithm can do for us. As affirmed by Fig. 3, a minimal representation usually

[3] Since SWP paths are not guaranteed to be SPR [15], we always substituted the corresponding shortest path representation.

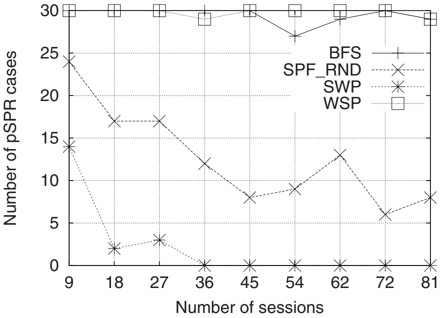


Fig. 4. Number of cases the minimal SPR was perfect

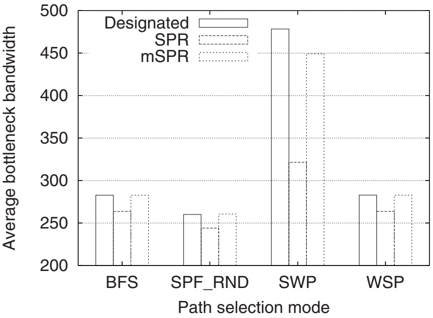


Fig. 5. Average bottleneck bandwidth of the paths

contains only a few superfluous paths up to the point that, except for SWP, it becomes almost perfect in most of the cases.

This observation is further confirmed by Fig. 4, which, as the function of the session number, shows the number of cases out of the total 30 simulations when the minimal representation turned out to be perfect as well. Observe that BFS and WSP paths are almost always pSPR. However, it seems that it is completely hopeless to expect a SWP path set to be pSPR, especially as the number of sessions grows close to the range of the number of nodes in the network.

Finally, we compared the average bottleneck bandwidth of the paths in the designated path set and its plain and minimal representations (see Fig. 5). While this choice obviously omits the interference amongst the sessions, the average bottleneck bandwidth is indeed a good measure of the transmission capacity that is made available by the network for the sessions. On the one hand, SWP is clearly superior in this regard by providing almost twice as much capacity as the other path selection schemes. On the other hand, sharpening the representation apparently improves the capacity of the paths in the representation (by one and a half times in the case of SWP). Our results indicate that in smaller networks the SWP algorithm combined with shortest path routing constitutes a really promising traffic engineering platform. Not just that SWP paths can be mapped quite accurately to shortest paths in this case but, in addition, these paths usually provide an abundance of capacity at the same time.

6 Conclusions

OSPF TE holds tremendous potential to optimize the performance of legacy IP networks. Wang *et al.* showed that whatever path set the traffic engineer assigns for the traffic instances, it is either immediately shortest path representable or otherwise, it can be reduced to a shortest path representable one. In this paper, however, we have shown that this result alone is not sufficient to warrant optimal performance of OSPF networks, because the process of mapping paths to shortest paths is highly prone to ambiguity. We have supplied both theoretical and empirical evidence that the concept of minimal representations is a remarkably useful one, or at least, it is much more useful than plain or perfect representations. A minimal representation constitutes a theoretical upper bound on the achievable precision and, moreover, it also contains perfect and plain representations as corner cases. In addition, using he mSPR algorithm, a minimal representation can always be computed in polynomial time by solving a series of linear programs.

References

1. B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communications Magazine*, vol. 40, pp. 118–124, Oct 2002.
2. A. Sridharan, C. Diot, and R. Guérin, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," in *Proceedings of INFOCOM 2003*, vol. 2, pp. 1167–1177, March 2003.
3. G. Rétvári and T. Cinkler, "Practical OSPF traffic engineering," *IEEE Communications Letters*, vol. 8, pp. 689–691, Nov 2004.
4. J. Moy, "OSPF Version 2." RFC 2328, April 1998.
5. Y. Wang, Z. Wang, and L. Zhang, "Internet traffic engineering without full-mesh overlaying," in *Proceedings of INFOCOM 2001*, vol. 1, pp. 565–571, April 2001.
6. G. Rétvári, R. Szabó, and J. J. Bíró, "On the representability of arbitrary path sets as shortest paths: Theory, algorithms, and complexity," in *Lecture Notes in Computer Science: Proceedings of the Third International IFIP-TC6 Networking Conference, Athens, Greece*, pp. 1180–1191, May 2004.
7. R. Guerin, A. Orda, and D. Williams, "QoS routing mechanisms and OSPF extensions." IETF RFC 2676, 1999.
8. G. Rétvári, T. Cinkler, and J. J. Bíró, "Notes on shortest path representation," Tech. Rep. 2005-001, Feb 2005. available on-line: `http://qosip.tmit.bme.hu/~retvari/publications/SPR_tech_rep.pdf`.
9. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs, NJ, 1993.
10. J. L. Kennington, "A survey of linear cost multicommodity network flows," *Operations Research*, vol. 26, no. 2, pp. 209–236, 1978.
11. M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows.* John Wiley & Sons, January 1990.
12. The GLPK project: `http://www.gnu.org/software/glpk/glpk.html`.
13. A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Universal topology generation from a user's perspective," Tech. Rep. 2001-003, Jan 2001.
14. Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal of Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.
15. J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet," in *INFOCOM*, pp. 727–735, 2001.

Achieving Bursty Traffic Guarantees by Integrating Traffic Engineering and Buffer Management Tools

Miriam Allalouf and Yuval Shavitt

School of Electrical Engineering,
Tel Aviv University
{miriama, shavitt}@eng.tau.ac.il

Abstract. Traffic engineering tools are applied to design a set of paths, e.g., using MPLS, in the network in order to achieve global network utilization. Usually, paths are guaranteed long-term traffic rates, while the short-term rates of bursty traffic are not guaranteed. The resource allocation scheme, suggested in this paper, handles bursts based on maximal *traffic volume allocation* (termed *TVAfB*) instead of a single maximal or sustained rate allocation. This translates to better SLAs to the network customers, namely SLAs with higher traffic peaks, that guarantees burst non-dropping. Given a set of paths and bandwidth allocation along them, the suggested algorithm finds a special collection of bottleneck links, which we term the *first cut*, as the optimal buffering location for bursts. In these locations, the buffers act as an additional resource to improve the network short-term behavior, allowing traffic to take advantage of the under-used resources at the links that precede and follow the bottleneck links. The algorithm was implemented in MATLAB. The resulted provisioning parameters were simulated using NS-2 to demonstrate the effectiveness of the proposed scheme.

1 Introduction

The latest Internet QoS (Quality of Service) design trends combine two approaches: DiffServ and MPLS. The first is based on reducing the computation complexity in core routers and on locating QoS entities such as policing and metering at the network edges. The DiffServ approach is based on per-hop QoS handling. In order to achieve global QoS guarantees or global profit gain, TE (Traffic engineering) tools are applied to design a connection-oriented network, e.g., using MPLS. In particular, QoS routing, where routes are assigned according to the service requirements, is an essential part to the end-to-end guarantees. Usually, the guarantees are applicable for long-term traffic rates, whereas the short-term rates of bursty traffic are not handled or guaranteed. This paper suggests a per-aggregate resource allocation algorithm that takes into account average traffic rates and also absorbs traffic bursts.

We consider as input a connection-oriented network where topology and directional link capacities are known. A typical rate demand of the network customer may represent aggregates of connections (e.g., TCP), such as client traffic (university campus, business client, client ISP), ATM VPs, or MPLS tunnels, and will be expressed by average or maximum required rate. The attitude of our resource allocation concept is to offer the network customers better SLAs with higher traffic peak rates that guarantees bursty traffic. It is a fast off-line algorithm that is performed during the network design phase.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 63–74, 2006.

Our resource allocation algorithm has two stages. In the first stage it seeks any QoS routing or bandwidth allocation algorithm that saturates the networks, such as maximum flow or max-min fair allocation [1]. Such algorithms use long-term average traffic demands as input, and allocate bandwidth using a signle rate parameter. In the second stage, we use buffers at specific locations for the short term traffic management, using the output of the long term TE algorithm. Note that we are not proposing to change the hardware whenever the demands are changed. All the routers will have their initial buffering resources, but our algorithms will use them optimally according to topology and demands analysis. These buffer analysis will determine the required flow regulation parameters at the edges of the network in order to enforce that traffic adheres to its designated maximal rate, while still isolating flows from each other. Specifically, we push the burst treatment to a point we term the *first cut*, which is an optimally selected set of bottleneck links. A burst is allowed to proceed unshaped until the destination, given the bottleneck link is not congested. In case of congestion the traffic is shaped at the *first cut* to the highest possible rate which guarantees the burst will not interfere with other flow traffic. Anyhow, the adjusted rate is never lower than the average rate determined by the long-term TE algorithm. Our algorithm determines provisioning parameters for the policy and regulation entities that are located at the edges of the network.

There are various methods for deterministic bandwidth allocation where the bandwidth is allocated using a single parameter, the maximal rate or the sustained rate parameter. The solutions of the different variants of the multi-commodity flow (MCF) problem for traffic engineering can be viewed as a long-term rate allocation method. Nichols *et al.* [2] describe two allocation methods for the DiffServ framework. The 'Premium service' is where the traffic is shaped at network edges. It provides the maximal permitted rate allocation contracts to its users, and it smoothes the jitter, provides certain delays, and guarantees peak rate flows. The 'Assured service' relies on statistical guarantees.

Other deterministic rate guarantees that consider the short-term rates [3, 4, 5, 6] were achieved by either the worst-case bounds on network internal buffer overflow or by end-to-end delays in the network. The rates of these traffic envelopes are not tight since they consider the worst-case bounds. A different line of research suggests statistical allocation guarantees. Christin *et al.* [7] examined the per-hop behavior of various real time streams having different constraints (such as delay or loss rate). Liebeherr [8] discusses different resource allocations and scheduling methods for the provision of delay sensitive video streams. Another approach is to allocate bandwidth according to an effective rate that takes into account statistical multiplexing between the burstiness of the flows [8, 9, 10]. Biton and Orda [11] provide QoS guarantees by coupling the scheduling mechanism and the routing schemes.

The resource allocation algorithm we propose in this paper reserves bandwidth according to the amount of traffic sent during a time interval (termed **TVAfB**, *maximal traffic volume allocation*) and not according to a single strict rate allocation (termed **MRA** in this work) used in previous suggestions.

The **TVAfB** cascading algorithm improves the state-of-the-art of service allocation and provisioning in a few ways. It allows bursty traffic to better exploit the existing network resources. It can also exploit the statistical multiplexing gain and still provides deterministic bandwidth and delay guarantees. For example, a burst that belongs to a flow that has only one bottleneck link that finds no congestion at this link can be transmitted further without any delay. In case of a higher load, but still below capacity, it flows in a higher rate than its sustained rate with no loss danger. Only during periods of congestion the burst is shaped to its fair share. The novelty of this approach lies in our dealing with bursty traffic guarantees and the fact that it employs the buffer as an additional resource in traffic engineering design.

Further, our algorithm can lead to higher parameters assigned for policing and regulation without being restricted to any specific policy method. The mathematical derivations we present in this work concentrate on the case where traffic is policed at the edges using token buckets. However, the notation of first cut is important and can be used for other regulation scenarios, as well. Section 2 presents the problem. Section 3 outlines the two-stage algorithm where section 4 details the second algorithm. Section 5 describes the simulation results and evaluates this proposition.

2 Problem Presentation

The algorithm considers a connection-oriented network where topology and directional link capacities are known. The set of paths are set optimally using any bandwidth allocation criterion chosen by the network administrator. We model the network as a general directed graph where each arc label represents link capacity. The traffic flow is assumed to be bursty, though the peering networks cannot explicitly express the burstiness characteristics. It is regulated by token buckets at the edge nodes. The token bucket parameters we seek per customer demand are token rate and bucket size. The regulation using these parameters determines the committed rate, the peak rates and the maximum burst size per path (CIR, PIR, and CBS). Our goal is to set the SLA regulation parameters in order to maximize the burstiness each flow is allowed, while at the same time not dropping packets by optimally use buffers along the routes. We will show that it increases bandwidth utilization for this type of traffic compared to the maximal rate allocation (**MRA**) that is usually used for long-term guarantees. Our algorithm shows that for many scenarios, there are paths with only one bottleneck link per path. In these cases, if buffers are allocated in this set of bottleneck locations, higher rate traffic per-path can be allowed to enter the network.

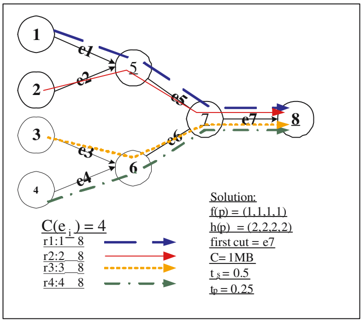


Fig. 1. Example 1

To illustrates the problem, Figure 1 depicts a simple directed network with 4 unidirectional paths. There are 4 different clients each with a demand of 1Mbps as depicted. All link capacities are 4Mbps. Thus, the bandwidth reservation is 4Mbps on link $e7$, 2Mbps on links $e5$ and $e6$, and 1Mbps on links $e1$, $e2$, $e3$, and $e4$, respectively. It is maximally allocated because link $e7$ is saturated. If a burst with peak rate of 2Mbps is sent along path $r1$, the packets exceeding 1Mbps will be dropped, though links $e1$ and $e5$ are not fully used. The rational behind our approach is to exploit links $e1$ or/and $e5$ capacity limits and still guarantee the traffic at the bottleneck, which in this case is link $e7$. By using another resource we can define extended allocation using more parameters, increase the usage of the under used links, and assign more flexible contracts. A 1Mbit buffer at the output port of node 7 to link $e7$ enables an agreement of 2Mbps peak rate, 1Mbps sustained rate and maximum burst time of 0.25 second for each path. The burst size for each path can grow as high as 2Mbit for a period of 0.25 seconds providing it is followed by a silence period of 0.25 seconds. Now consider an underload situation where only one client transmits bursty traffic of 2Mbps peak rate. This stream will be transmitted without any buffering delay all the way. Otherwise, if all the sources transmit using their peak rate, the buffer at node 7 will shape (using any GPS-compliant scheduler) the traffic per path to the sustained rate.

3 Algorithmic Solution

Below is an outline of the algorithm that achieves deterministic guarantees for bursty traffic. The algorithm is based on a few algorithms activated in cascade.

3.1 Solution Outline

1. 1^{st} **stage - Routing and Average Rate Allocation**: Find, using LP (Linear Program) formulation and solver, the QoS routing that identifies maximum flow (or other criterion) allocation of the bandwidth. The output is the set of paths and the net flow that is assigned per path. This stage is described in Section 3.2.
2. 2^{nd} **stage - TVAfB cascading algorithm - Traffic Volume Allocation for Bursts**:
   (a) Find a special set of bottleneck links, termed the *first cut* (Section 4.1).
   (b) Indicate which buffers at the *first cut* enable us to increase the rate at the edges.
   (c) Calculate the permitted peak rate over each path taking into account all the arcs not included in the *first cut* for each path. Again, we use LP solver over the residual graph 'before' and 'after' the first cut (Details in Section 4.2).
   (d) Based on the previous calculations, decide for each path whether it can gain additional burstiness using buffering. If yes:
      – Analyze buffer behavior at the bottleneck link, in case of congestion (4.3).
      – Set a contract (SLA) per-path (Section 4.4).

3.2 1^{st} stage: Long-Term Routing and Bandwidth Allocation

This stage specifies a set of paths in the network, and allocates them bandwidth. TE tools are used to choose paths between a given set of ingress-egress pairs. Any resource allocation criterion can be used, in order to saturate the network.

In this paper we are particularly considering the Maximum Multi-commodity Flow (MCF) problem. The input to this problem is the network topology, the directional links capacities, and a list of ingress-egress pair (clients). It finds the maximum of the total net flows over all commodities (e.g, paths), the routing to be used between each pair, and the net flow per each path. This problem can be solved using LP solver in a polynomial number of steps. We specifically consider this problem since it achieves network saturation and leaves minimal excess capacities. Other routing algorithms that allocate bandwidth and saturate the network can also fit this framework. In [1] we suggested bandwidth allocation method according to the max-min fair criteria that can be used for the *TVAfB* algorithm.

4 2^{nd} Stage: TVAfB Cascading Algorithm - Traffic Volume Allocation for Bursts

4.1 The Bottleneck Links for Buffering Analysis

The 1^{st} stage solution found the set of paths between (s_i, t_i)-pairs and a per path net flow $f(P)$ in the graph $G(V, A)$. Based upon the routing found previously this subsection will find the strategic location for the buffers, which is defined below as *first cut*. First, we will define a few terms.

Definition 1. *A link a is saturated, denoted: $sat(a) = 1$ if it is assigned bandwidth equal to its capacity. Otherwise it is not saturated which is denoted $sat(a) = 0$.*

Definition 2. *a_i is the fbn link of a path $p = (a_1, a_2, \ldots), a_j \in A$, if $i = \min\{j|sat(a_j)\}$*

Definition 3. *A* first cut *is the set of the first bottleneck links ($fbns$).*

Definition 4. *Given a graph $G(V, A)$ and a set $(s_i, t_i), \forall i = 1..K$ of source-terminal pairs, a cut M of the graph is a subset $M \subset A$ such that the subgraph $G‘ = (V, A \setminus M)$ has no $s_i \rightarrow t_i$ path, $\forall i = 1..K$.*

Using the above definitions we can state the main construction of this subsection.
The *first cut* properties

1. Each path has exactly one fbn link. The number of fbn links $\leq$ the paths.
2. For each path, the links that are prior to its first bottleneck link are under-used.
3. Each *first cut* link can be saturated by flows that this link is their fbn link and by other flows that already met their fbn link before (discussed in 4.2).
4. The *first cut* is a cut of the graph. If we delete the arcs of the first cut no traffic will flow (The proof can be found in [12]). Thus, we can use it as the location for absorbing the peak rates of the bursts.

4.2 Peak Rates Calculations

The *Traffic Volume* allocation assigns peak rate $h(p)$ per path p on top of the sustained rate, $f(p)$, which was found in the 1^{st} stage. The lower bound for each $h(p)$ is $f(p)$. The goal of this work is to enable flow transmission over a predefined path using its peak rate when the buffer is used only in case of congestion. Therefore, the peak rates calculation is derived out of the excess bandwidth of the links, which are not saturated, and is divided among all the paths flowing through them.

This subsection calculates the possible peak rates per path in each first bottleneck link (fbn) subject to capacities constraints of all the preceding and following arcs over this path. For this purpose we use the same TE algorithm used in the first stage over the residual graph arcs that reside 'before' and 'after the first cut. The specific TE algorithm (maximum flow, max-min fair, etc.) also determines how the excess bandwidth will be divided among the paths.

The construction of the 'before' and 'after' residual graph is as follows. According to property 2 of the *first cut*, all the links of path p prior to its fbn are under used and can accommodate higher rates than the sustained rate $f(P)$. However, property 3 is more complicated. Consider a link fbn_1 (belonging to the *first cut*) and a set of paths that are traversing it. Note that fbn_1 may not be the first bottleneck link for some of the paths that traverse it. Assume a path p_i which passes through the saturated links fbn_2 and fbn_1 in this order. By definition only fbn_2 is p_i's first bottleneck link. However, peak rates calculation, residual graph construction and buffer management vary if p_i has more than one bottleneck link. Essentially, this variation arises due to the need to allocate these peak rates along the arcs that lay between the bottleneck links (fbn_2 and fbn_1).

We developed two algorithms. The first, algorithm A, saves buffering resources by allowing burstiness (some peak rate) only for paths that traverse a single saturated link. The second, algorithm B, enables burstiness also for paths that traverse multiple saturated links, but requires more buffering resources. In both algorithms, shaping of the peak rate to the sustained rate is performed only when congestion occurs, otherwise, the flow's peak rate is allowed.

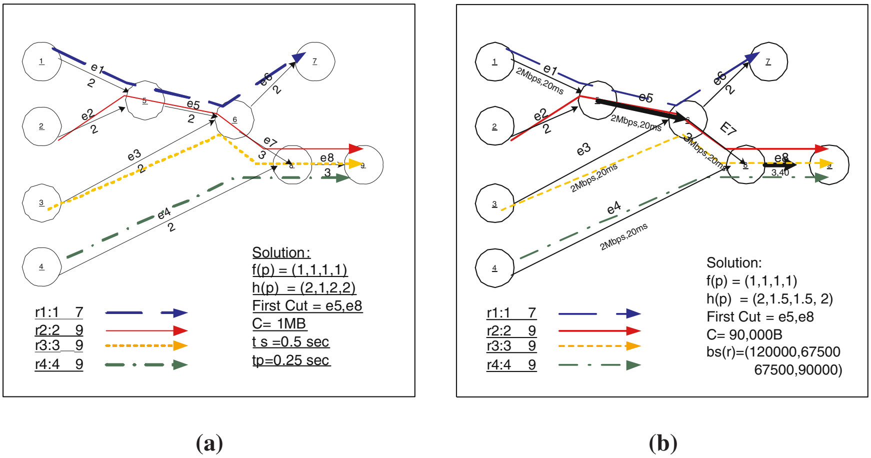
Peak Rate Calculation Algorithm A: Enabling Burst Flow only for Single-fbn Paths. The first algorithm benefits paths that traverse a single fbn link whose other links (not in the **first cut**) are under used. The excess bandwidth in the under-used links is divided among these paths, which permits a possible peak rate per path. Not every topology and demand flow can benefit from this algorithm, though the algorithm can check its usefulness. Section 5 discusses briefly the topologies that are likely to be beneficial by the algorithm. The traffic flow is controlled at the ingress, using the peak rate. Other traffic flows are controlled using the sustained rate. In case of congestion, buffers at the first cut will be used to shape the peak rate to a lower rate (but not lower than the sustained rate).

The input for this algorithm is the graph $G(V, A)$; its arc capacities; set of paths over G and the assigned net flows over them and the *first cut* arcs. The algorithm finds $h(P)$, the permitted peak rate per path in two steps. The first step constructs a sub-graph $G^-(V, A^-)$ (see in Figure 2). The second step applies the TE algorithm used in the 1^{st} stage over A^- and identifies the highest possible rates over the paths subject to A^- capacity constraints.

Consider the example in Figure 3(a), where the arc capacities of links $e1 - e6$ is 2Mbps and of links $e7 - e8$ is 3Mbps. The optimal bandwidth assignment per-path, calculated by the *first-stage TE* algorithm is 1Mbps. We consider this rate to be the sustained rate. The *first cut* consists of the links $e5$ and $e8$. Paths $r2, r3$, and $r4$ are traversing arc $e8$. Note that $fbn(r3) = fbn(r4) = e8$ but $fbn(r2) = fbn(r1) = e5$. Paths $r1, r3$, and $r4$ have only one *fbn* link, thus, their rate can be increased. Path $r2$, however, is excluded from the set of the beneficial paths because it has two bottleneck links and can not have burstiness. A^- contains links $e1$ and $e6$ (that precedes and follows $e5$ respectively), $e3$, $e4$, and $e7$ (that are prior to $e8$). The residual capacity of $e1$, $e3$ and $e4$ in A^- is 1 (originally was 2) and the capacity of $e7$ is 1 (originally 3). Buffer located

```
**Constructing set of links** $A^-$
Set $FPATHS$ to be the set of all input paths (from TE stage), $NEWFPATHS = FPATHS$
**for** each bottleneck link $a$ in 'first cut':**do**
    Set $FP(a)$ to be all the paths passing through $a$
    **for** $f_i \in FP(a)$ **do** /* Consider only paths with single $fbn$ */
        **if** $a = fbn(f_i)$ and $\forall a_f \in f_i, a_f \neq a, a_f \notin firstcut - a$ **then**
            **for each** $a_f \in f_i, a \neq a_f$ **do** $A^- = A^- \bigcup a_f$
        **else** $NEWFPATHS = NEWFPATHS - f_i$
/* Get the residual graph : for the excess rates calculation */
**for each** $f_i \in FPATHS$ **do, for each** $a_f \in f_i$ **do** $c(a_f) = c(a_f) - f(f_i)$
```

Fig. 2. Algorithm A G^- construction: selecting links for the peak rate



(a) **(b)**

Fig. 3. Results of algorithms A and B for a network with various arc capacities

at the *first cut* links $e5$ and $e8$ absorbs the sum of the peak rates of the traversing paths (which is (2,1,2,2) for paths 1,2,3 and 4). The derivation of the maximum peak period per path that is allowed subject to the buffer size and the calculated peak rate is described in subsection 4.3. In this algorithm, each flow peak rate is only considered once in the buffers calculation, at its first bottleneck link. This means that our usage of the buffering resources is minimal and is not sensitive to whether the first cut is the minimum cut or what is the number of the links of the first cut. The maximal peak rate, R_p, that can be handled at each one of the first cut links is **not the sum** of the peak rates of the paths that traverses it, but is given by $\forall a \in A^-, R_p^a = \sum_{\text{already shaped paths}} f(p) + \sum_{a \text{ is their fbn}} h(p)$.

Peak Rate Calculation - Algorithm B: Enabling Bursts Flow for all the Paths, with more buffers. This algorithm enables peak rates assignment also to paths with more than one *fbn* link though this requires more buffering resources. As in algorithm A, we build a new sub graph $G^-(V, A^-)$ and apply the same TE algorithm on G^- to find $h(P)$, the per-path permitted peak rate. A^- consists of all the links except the *first cut* links. In this algorithm, assuming there is no congestion in the network, a flow of a path that traverses more than one bottleneck link can reach the second bottleneck link with a higher rate than its sustained rate. Portion of the buffer in this *fbn* has to be assigned to guarantee the higher rates. Consequently, more buffering resources should be added at each first cut link to accommodate the peak rates.

Figure 3(b) shows algorithm B execution on the same graph used in Figure 3(a). The rate of path $r2$ can be increased even though it has two *fbn* links, and its peak rate is calculated using arcs $e2$ and $e7$. There will be 2 buffers: one located at node 5 towards $e5$ to treat bursts from routes $r1$ and $r2$ and the other is located at node 8 towards $e8$ to treat the bursts of routes 2,3 and 4. Assuming locating buffers of size 90,000 bytes at the output ports of nodes 5 and 8 towards links $e5$ and $e8$. The sustained rates are (1Mbps, 1Mbps, 1Mbps, 1Mbps), peak rates are (2Mbps, 1.5Mbps, 1.5Mbps, 2Mbps), and the sizes of the token buckets are (120,000, 67,500, 67,500, 90,000) bytes for routes (1, 2, 3, 4), respectively. The details of this calculations can be found in subsections 4.3 and 4.4. Note that the *fbn* link $e5$ allows a burst size of $90,000$ for path $r2$ but this burst size was decreased by the fbn $e8$ upper bound. As in the previous algorithm, in case where a path cannot gain a peak rate that is higher than its sustained rate, it will be policed to its sustained rate at the ingress. Otherwise, the peak rate will be used.

4.3 Buffer Management Analysis at the *First Cut*

The buffers, located at the **first cut**, are used for holding the bursts that may arrive with a maximal rate of $h(p)$ for any path p. The buffer sizes are determined by the peak rates calculated in 4.2. Given the shaping capabilities at the first cut, we can calculate the possible traffic envelopes at the first cut. The way we handle the traffic at the first cut affects the control parameters of the traffic at the ingress nodes. Many previous papers estimated the bounds on the size of traffic envelopes at the core based on the traffic pattern at the source nodes. Since our calculations are derived from the TE routing stage, we are able to set regulation rules at the ingress. Specifically, we assume the incoming flows are regulated per path using token buckets at their source node. We derive the per-path token bucket parameters (i.e., peak rate, sustained rate, and burst size) from the first cut buffer analysis. Figure 4 describes the node's functionalities with buffer capacity C, link output rate, R_{out}, peak rate of arriving traffic, $R_{peak,in}$, and a peak interval, t_p. The transmission rate of the outgoing traffic is bounded by the link output rate, R_{out}. If the rate of the offered traffic is $R_{in} \leq R_{out}$, a queue will not build up. In case of bursty traffic the buffer is used for storing the incoming packets which are smoothed by the transmission rate. The most extreme case is an On-Off streams in an interval t_s, which are composed of peak rate $R_{peak,in}$ for the burst duration t_p followed by a silence period of length $t_s - t_p$. The longest period of time t_p that a burst can be sent, given, $R_{peak,in}$, R_{out} and C is expressed by:

$$t_p = C/(R_{peak,in} - R_{out}) \tag{1}$$

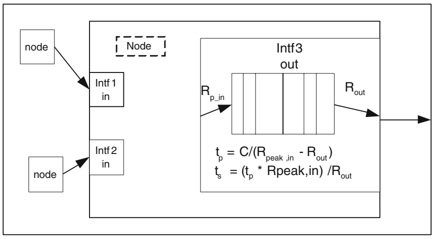


Fig. 4. Buffer management at the output port

The minimal length of the interval t_s can be derived by equating the amount of incoming and outgoing data:

$$t_s = R_{in} \cdot t_p / R_{out} \tag{2}$$

Alternatively, we require that the generated amount of data v in the interval t_s: $v \leq R_{out} \cdot t_s$. The maximum delay at a node is given by the emptying time of a full buffer C/R_{out}. A general definition of v will be to integrate the arrival rate, given $g(t) \overset{\text{def}}{=} R_{in}(t)$: $\int_{t-t_s}^{t} g(t)d(t) \leq (R_{out} \cdot t_s)$ where t_s is calculated from using Eq. 1 and Eq. 2. We have shown that if the above parameters on the arriving traffic are kept, the traffic is guaranteed to be conforming. Next we will prove the correctness of traffic envelope bounds. Consider streams $i = 1, 2, \ldots$ with peak rates $h(p_i)$, sustained rates $f(p_i)$, and $\int_{t-t_s}^{t} g_i(t)d(t) \leq f(p_i) \cdot t_s$. The following Lemma states the conditions for conformance.

Lemma 1. ***Assuming*** *outgoing link rate R_{out}, permitted peak rate $R_{peak,in}$, buffer capacity C, time t_s and m input traffic streams.* ***If*** *(1) $\sum_{i=1}^{m} h(p_i) \leq R_{peak,in}$, (2) $\sum_{i=1}^{m} f(p_i) \leq R_{out}$ and (3) $\forall i = 1, \ldots, m \int_{t-t_s}^{t} g_i(t)d(t) \leq f(p_i) \cdot t_s = h(p_i) \cdot t_p$* ***holds, then*** *the total volume $v \leq R_{out} \cdot t_s$.*

The proof can be found in [12]. The sum of burst sizes of the input streams equals to the maximal permitted g(t) so there will be no data loss.

4.4 Setting Per-Path Token Bucket Parameters

The following subsection describes the algorithm that assigns each path with its token bucket parameters: the token fill rate and the bucket size. The token fill rate governs the per path sustained rate and the bucket size is calculated by the maximal burst time interval t_p multiplied by the peak rate. We derive these parameters by traversing each first cut arc. We assume all first cut links have the same buffer size C. By applying these parameters to the token bucket at the ingress of this path, the traffic is assured to be conforming.

- **Perform** for each $a^k \in A^-$ with outgoing rate R_{out}^k

  1. For each incoming path p_i: $h(p_i) = \begin{cases} f(p_i) & \text{/*cannot increase its rate*/} \\ h(p_i) & \text{/*otherwise */} \end{cases}$
  2. Set $R_{peak,in}^k$ to be the incoming peak rate of a^k, $R_{peak,in}^k = \sum_{path\ i \in a^k} h(p_i)$.
  3. set t_p^k to be the maximal burst interval for arc a^k using Eq. 1, C, R_{out}^k, and $R_{peak,in}^k$.

Table 1. provisioning parameters can be systems wide (the only one here is buffer size), per path, or per node interface

Parameters	**Per-**fbn	**Per-Path**
Buffer size, Same for all $fbns$	C	
R_{out}	The fbn interface link rate	
$R_{peak,in}$	The sum of peak rates per-path ($\sum h(p_i)$)	calculated per fbn in subsection 4.2
t_p	Calculated using C, R_{out} and $R_{peak,in}$ (Eq. 1)	The minimum over all *first cut* links it traverse
Burst size	$R_{peak,in} \cdot t_p$	$h(p_i) \cdot t_p$

4. Apply to all the paths of a^k (that a^k is their fbn) the values $f(p_i), h(p_i)$ and t_p^k. Set the token bucket contract to be: token rate $= f(p_i)$ and $bs = h(p_i) \cdot t_p^k$

Table 1 summarizes the parameters this system needs for provisioning and the order of their derivation. All the stages of the algorithms were implemented using MATLAB.

5 Simulation Results and Evaluation

Simulations. In order to evaluate the gain from our algorithm, we applied both allocation methods, **TVAfB** and the the **MRA** using the NS-2 simulator and the example in Figure 3(b). The four aggregates in the example are composed of 10 TCP[1] connections (each with maximal congestion window size of 100), and use different paths, $r1, .., r4$. Each TCP connection transfers a file of 2MByte.

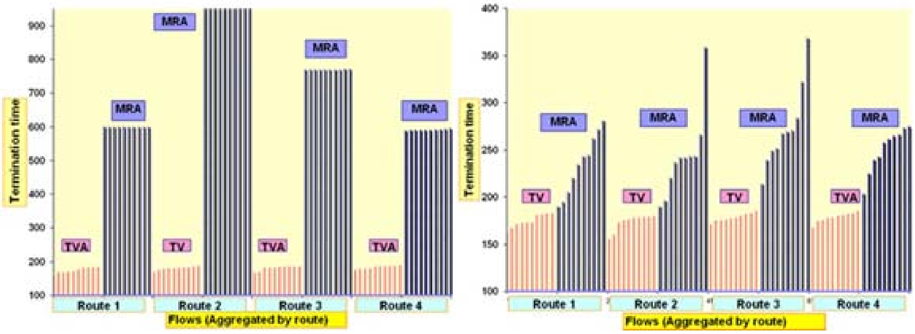


Fig. 5. The height of a per-connection vertical bar indicates the termination time of the appropriate TCP flow. Every ten bars are grouped by aggregate, for *TVA* (Bars group:1,3,5,7) and *MRA* (Bars group:2,4,6,8).

The regulation entities (token buckets) that are located at the ingress nodes, $1, 2, 3$, and 4, perform policing and metering for the arriving aggregates, namely all the 10 TCP connections are policed together. The *MRA* only allows packets that arrive within the maximal rate, $1Mbps$ in this example. We set the tokens fill-rate to be $1Mbps$ and the bucket size to be 1000B (equals to the size of 2 packets). The token bucket parameters for the *Traffic Volume Allocation* (TVA) are the values that are calculated in Section 4.4 and presented in Figure 3(b). In both methods, any 'out-of-profile' packet is dropped, though we allow bursts in the size of the token bucket. Further, we locate weighted queues of 186 packets (equals to 90,000 Bytes) at the output ports of nodes 5 and 8 towards arcs $e5$ and $e8$. We use propagation delay of 20ms for all the links in each direction, except for link $e8$ whose propagation delay is 40ms.

The simulation measures the time it takes for each connection to transmit the 2Mbyte file. We compare the per-aggregate average termination time, computed over all the

[1] TCP was selected due to its bursty nature and its prevalence in today Internet. This enforces us further to discuss the TCP congestion control in the context of our work.

connections within each aggregate, and the number of the dropped packets per-aggregate. Figure 5(a) depicts the simulation termination results for the two allocation methods for all the connections. Clearly, *TVA* gained a $2.5 - 4.5$ speedup in the file transfer time. The reason for this is the higher number of conforming packets, and thus less drops. Indeed, for *TVA* the average drop rate is 2.5%-6$\%$, while for *MRA* it is 16.7%[2]. The file transfer times for the *MRA* are much longer than *TVA* because of the huge 'out-of-profile' dropping, which causes TCP timeouts. Running the same example but with 1/10th of the propagation delay over all the links (see Figure 5(b)) decreases the termination times that are achieved by the *MRA* since it decreases the time the slow-start phase requires to ramp up. It does not affects *TVA* performance since it spends its time in congestion avoidance (due to the small percent of packet drops) and the policer allows it to transmit enough packets, such that it start receiving acknowledgements before it exhausts its window. To further study our algorithm performance, we looked at more scenarios where the loads over the different routes are not even such that the bottleneck link $e8$ is under used. All the TCP connections that participated in a non-even scenario increased their rates related to the even-load scenario[3].

A common real-world architecture that can benefit from using the *TVAfB* algorithm is an access or a metro network. In a common metro architecture, a set of paths from the clients (modem pools, T1 lines, etc.) forms a tree towards the ISP Internet gateway. The link capacities in this network are the same due to a homogeneous usage of technology, e.g., 1Gbps Ethernet. Thus, the link to the gateway router becomes a bottleneck and an fbn in the *TVAfB* algorithm. This link capacity, $1Gbps$, is shared by the sustained rates of all the paths. Obviously all the preceding links have an excess bandwidth that can be added to the rate of the paths. Furthermore, the needed buffering resource in the gateway router are modest[4].

6 Concluding Remarks

The solutions presented in this paper can be used by network administrators as a design tool. The algorithm assumes the knowledge of the traffic rate demands across the network and the ability to lay a set of fixed routing paths. It can be performed as often as any *keep-alive* algorithm in a connection-oriented network. Beside the fact that all the algorithms runs in a polynomial number of steps, we verified the practicality by examining issues such as required buffer size and shaping algorithms. It is a fast and easy-to-deploy algorithm that can be used over one or more network domains, in order to find the bottleneck links, buffering needs, and SLA parameters.

Acknowledgments: We thank Danny Dolev for many helpful discussions.

[2] Note that the *TVA* transfer time is only 50% higher than TCP theoretical achievable rate.

[3] This framework can use a model that sizes the buffer of a bottleneck link considering the parameters of the TCP sources [13].

[4] Assuming this router has 16 1Gbps input-ports which are aggregated into one 1Gbps output link, and a burst period of 1ms, the cumulative burst size $BS = (1Gbps \cdot 16 - 1Gbps) \cdot 1ms = 15,000,000bits \simeq 2MB$, meaning only $2MB$ to be shaped, in case of congestion.

References

[1] Allalouf, M., Shavitt, Y.: Centralized and distributed approximation algorithms for routing and weighted max-min fair bandwidth allocation. (In: IEEE HPSR'2005)

[2] Nichols, K., Jacobson, V., Zhang, L.: A two-bit differentiated services architecture for the internet – RFC no. 2638. Internet RFC, Internet Engineering Task Force (1999)

[3] Cruz, R.L.: A calculus for network delay part II: Network analysis. IEEE Trans. on Information Theory **37** (1991) 132–141

[4] Parekh, A.K., Gallagher, R.G.: A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. IEEE/ACM Transactions on Networking **2** (1994) 137–150

[5] Georgiadis, L., Guérin, R., Peris, V., Sivarajan, K.N.: Efficient network QoS provisioning based on per node traffic shaping. IEEE/ACM Transactions on Networking **4** (1996)

[6] Yaron, O., Sidi, M.: Generalized processor sharing networks with exponentially bounded burstiness arrivals. In: INFOCOM (2). (1994) 628–634

[7] Christin, N., Liebeherr, J., Abdelzaher, T.: A quantitative assured forwarding service. In: IEEE INFOCOM 2002, New York, NY, USA (2002)

[8] Liebeherr, J.: A note on statistical multiplexing and scheduling in video networks at high data rates (2002)

[9] Clark, D., Lehr, W., Liu, I.: Provisioning for bursty internet traffic: Implications for industry structure. In: MIT ITC Workshop on Internet Quality of Service. (1999)

[10] Guerin, R., Ahmadi, H., Naghshineh, M.: Equivalent bandwidth and its application to bandwidth allocation in high-speed networks. IEEE Journal on Selected Areas in Communications **9** (1991) 968–981

[11] Biton, E., Orda, A.: Qos provision and routing with stochastic guarantees. (In: QShine'04)

[12] Allalouf, M., Shavitt, Y.: Achieving bursty traffic guarantees by integrating traffic engineering and buffer management tools. Technical report, Dept. of EE, Tel Aviv University (2005) EES2005-50.

[13] Dhamdhere, A., Jiang, H., Dovrolis, C.: Buffer sizing for congested internet links. (In: INFOCOM'05)

How Well Do Traffic Engineering Objective Functions Meet TE Requirements?

Simon Balon*, Fabian Skivée, and Guy Leduc

Research Unit in Networking,
EECS Department- University of Liège,
Institut Montefiore, B28 - B-4000 Liège - Belgium
{balon, skivee, leduc}@run.montefiore.ulg.ac.be

Abstract. We compare and evaluate how well-known and novel network-wide objective functions for Traffic Engineering (TE) algorithms fulfil TE requirements. To compare the objective functions we model the TE problem as a linear program and solve it to optimality, thus finding for each objective function the best possible target of any heuristic TE algorithm. We show that all the objective functions are not equivalent and some are far better than others. Considering the preferences a network operator may have, we show which objective functions are adequate or not.

1 Introduction

We consider the traffic engineering routing problem. Given the topology of the network to be engineered and an estimate of the traffic matrix to be routed on it, the problem is to find a routing scheme that optimises the network, with the joint goal of good user performance and efficient use of network resources. The way classical algorithms fulfil this objective is not clear. Indeed, many algorithms try to optimise their home-made objective functions which are said (but not proven) to reflect traffic engineering objectives. The foundations of all these objective functions are related, but could lead to quite different results, as we see in our simulations.

Some in-depth reflection and comparison studies of traffic engineering objective functions are needed. In many research papers, the quality of a new traffic engineering algorithm is evaluated regarding one specific objective function. If the algorithm obtains a good score, it is considered as good. But this is only meaningful if the objective function really reflects the traffic engineering goals. Furthermore, when analysing published papers it is difficult to figure out if the merits of a given TE method is due to its objective function or its heuristic algorithm. To fill this gap, we provide an independent comparison of many objective functions found in the literature.

To compare all the different objective functions, we will minimise (or maximise) each of these functions on the same topology and traffic matrix and analyse

* S. Balon is a Research Fellow of the Belgian National Fund for the Scientific Research (F.N.R.S).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 75–86, 2006.

if the routing scheme we obtain really reflects general Traffic Engineering goals. One important point is that we have used some real topologies and real traffic matrices to run our tests, which is not the case of many research papers. The use of real data provides a real case study and an objective basis for comparison.

Section 2 presents Traffic Engineering goals and requirements. Section 3 introduces existing TE algorithms and related objective functions. We discuss the foundation of these objective functions and why they were introduced. In section 4, we construct LP (Linear Programming) models of these objective functions. These models are used to compare all the presented functions on different networks. Then we analyse the results of simulations, highlighting the merits and/or shortcomings of each objective function. Finally, section 5 concludes the paper.

2 Traffic Engineering Objectives

A network is modeled as a directed graph, $G = (N, A)$ whose nodes and arcs represent routers and links. Each arc has a capacity c_a. Traffic on the network is represented by a traffic matrix D that with every pair (s, t) of nodes associates the value of the traffic demand, i.e. the traffic that flows from node s to node t.

Basically, the graph G and the traffic matrix D are the inputs of the problem. A traffic engineering algorithm has to find *good* paths between each pair of source and destination nodes to route corresponding traffic flow. The definition of *good* paths is related to what we want to optimise on the network. Generally, a good set of paths will be one that optimises a pre-defined objective function.

Once the paths are chosen, we can associate with each arc a load l_a, which is the total load on the arc, i.e. the sum over all demands of the amount of traffic sent over a. The utilisation of a link a is $u_a = l_a/c_a$. The available bandwidth on link a is $ABW_a = c_a - l_a$.

Finally, we define θ_{st} as the maximum flow that can be sent from node s to node t in the residual network, i.e. when the whole traffic matrix is routed on the network.

2.1 Discussion on TE Objectives

Typically, on-line algorithms have different objectives than off-line ones. On-line schemes usually try to minimise the probability of blocking future requests, while off-line ones try to minimise the load or the utilisation of the links, or try to maximise available bandwidth. To some extent, minimising the link utilisation (which is a relative measure) tends to maximise the available bandwidth (which is an absolute measure) on the links, thus also reducing the blocking probability of future requests. Clearly, these objectives are closely related, but no solid basis exists to choose one among all.

We will consider TE metrics at three different levels, which are a link, an OD pair[1] and the network. We will present and justify the foundation of the TE metrics at each level. We will differentiate metrics whose goal is to improve the

[1] OD stands for Origin Destination.

quality of the network given the present traffic (e.g. minimise the delay) from metrics whose goal is to maximise the acceptance of future traffic on the residual network (e.g. maximise residual max-flow).

At the *link level*, we should minimise delay and utilisation. We should also maximise the available bandwidth on this link (which corresponds to the notion of residual max-flow for a link). The delay of a link is composed of three components: the propagation delay which is a constant value, the transmission delay (inversely proportional to the link capacity) and the queueing delay which increases with the link load. If we take the delay to be the average delay of an M/M/1 queue, the mean queueing + transmission delay of link a is given by $Delay_a = \frac{mean\_packet\_size}{c_a - l_a}$. For a M/M/1 queue, all the percentiles/quantiles are also proportional to this value. On high capacity links, this delay is significant only if the link load is approaching the link capacity. Figure 1 summarizes the relations between link parameters.

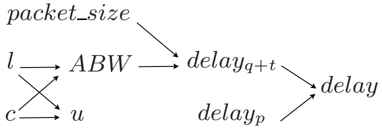


Fig. 1. Link parameters

At the level of an *OD pair* of nodes, we should minimise the path delay, i.e. the sum of the delays of all the links on the path. Minimising this delay can increase the quality of service perceived by the users of the network. We should also minimise the maximal link utilisation on the corresponding path. Indeed, the maximal link utilisation has a particular meaning. For example a maximal link utilisation (u_{max}) of 50% means that we can double all the traffic before having a link fully loaded (if we keep the same routing scheme), while a value of 20% means that we can multiply all the traffic by 5. In fact this value ($\frac{1}{u_{max}}$) is a lower bound because a change in the routing scheme may allow increasing this value. Finally, the residual max-flow between an OD pair of nodes should be maximised. Indeed, this value represents the maximal size of a future request that can be routed on the network between these nodes.

We have now some ideas of TE metrics to be optimised for a link or for an OD pair. But to be really useful in TE algorithms we have to generalise these concepts to the *whole network*. There are many ways to proceed. For example, considering link utilisations, one can minimise the maximal link utilisation (u_{max}) as for the OD level[2]. But the minimisation of the maximal link utilisation works poorly in some cases. Indeed if there is a real bottleneck in the network, i.e. a link whose utilisation cannot be decreased by changing the routing scheme, it is important

[2] We can prove that the routing scheme that achieves the minimal value of maximum link utilisation also provides the optimal value concerning the factor by which it is possible to multiply the current traffic matrix. In this case, this factor can be computed as $\frac{1}{u_{max}}$.

to minimise also the utilisation of other links. One way to proceed can be to minimise the mean link utilisation. Considering the delay to be minimised on the whole network, we can compute the mean link delay (each link being weighted by its load or not) or the mean path delay (each path being weighted by its corresponding traffic). The unweighted mean path delay seems less relevant to us. The following demonstration shows that the weighted mean path delay is equivalent to the weighted mean link delay. This demonstration highlights that it is possible to compute the mean path delay without path information. This is an important result from a computational point of view. Indeed it is less complex to compute a sum over all links than a sum over all paths of all possible OD pairs of nodes.

$$
\begin{aligned}
&MeanDelay = \tfrac{1}{\sum_{(s,t)} D(s,t)} \sum_{(s,t)} D(s,t) \sum_{a\in\mathcal{P}(s,t)} delay_a \\
&= \tfrac{1}{AllTr} \sum_{(s,t)} D(s,t) \sum_{a\in\mathcal{P}(s,t)} delay_a \\
&= \tfrac{1}{AllTr} \sum_{(s,t)} D(s,t) \sum_{a\in A} \delta_{a\in\mathcal{P}(s,t)} delay_a \\
&= \tfrac{1}{AllTr} \sum_{a\in A} delay_a (\sum_{(s,t)} D(s,t)\delta_{a\in\mathcal{P}(s,t)}) \\
&= \tfrac{1}{AllTr} \sum_{a\in A} l_a \times delay_a
\end{aligned}
$$

$\mathcal{P}(s,t)$ denotes the path from s to t[3], $\delta_{a\in\mathcal{P}(s,t)}$ is equal to one if link a belongs to $\mathcal{P}(s,t)$ and 0 otherwise. $AllTr$ denotes the sum of all traffics of the network $(\sum_{(s,t)} D(s,t))$. It is a constant for a given problem.

Considering max-flows (θ), it is possible to maximise the minimal residual max-flow. But as for the maximal link utilisation, the minimal max-flow can be blocked by a set of bottleneck links. So we should also maximise the sum of all max flows (instead of the min value). We could also associate with each max-flow a weight related to its corresponding traffic demand.

As it could be interesting to maximise the sum of residual max-flows, it could be interesting to maximise the sum of the available bandwidths over all the links of the network. However, we can notice that it is equivalent to minimising the sum of the loads over all the links of the network. Indeed, $max \sum_{a\in A} ABW_a = max \sum_{a\in A}(c_a - l_a) = max(\sum_{a\in A} c_a - \sum_{a\in A} l_a) \equiv min \sum_{a\in A} l_a$, as the sum of all the capacities of the network is invariant.

Table 1 presents a summary of TE metrics introduced in this section.

Table 1. TE metrics summary

	Metric characterising good current state	Metric characterising likely good future
Link$_{(a)}$	$Delay_a$	u_a, ABW_a
Path$_{(s,t)}$	$\sum_{a\in\mathcal{P}(s,t)} Delay_a$	θ_{st}, $max_{a\in\mathcal{P}(s,t)}u_a$
Network	$\frac{\sum_{a\in A} Delay_a}{\lvert A\rvert}$, $\frac{\sum_{a\in A} l_a \times Delay_a}{AllTr}$	$min_{(s,t)}\theta_{st}$, $max_{a\in A}u_a$ $\sum_{(s,t)} \theta_{st}$

[3] Here, we assume that there is only one path used from s to t, but the demonstration can be easily generalised if there are multiple paths.

2.2 How to Measure the Quality of a Solution?

In this section we present the TE metrics we will use to evaluate the quality of the routing solutions in the simulation section. As presented in the preceding subsection, it is clear that the maximum link utilisation (u_{max}) is a good TE metric. In addition, the mean link utilisation (u_{mean}), the 10^{th} percentile (u_{per10}), the minimal available bandwidth (ABW_{min}) and the mean load (l_{mean}) will be used. u_{per10} is defined so that 10% of the links have a utilisation over u_{per10}. We think that the weighted mean queueing + transmission delay of the network ($delay_{mean} = \frac{1}{AllTraffic} \sum_{a \in A} l_a \times \frac{packet\_size}{c_a - l_a}$) is also an important variable. We will also consider the minimum max flow ($\theta_{min} = Min_{(s,t)} \theta_{st}$) and total max flow ($\theta_{tot} = \sum_{(s,t)} \theta_{st}$) of the residual topology. The total max flow gives an idea of the throughput, i.e. which amount of traffic can be accepted on the residual network. This is not exactly the amount of bandwidth that can be routed on the residual network because all max-flows are computed independently of each other and thus all the flows are not in competition for the residual bandwidth. But this can still give a good idea of the residual throughput[4].

3 Presentation of Different Objective Functions

3.1 Fortz

In [1], B. Fortz et al. try to find an optimal set of IGP weights such that classical shortest path first algorithms taking these modified metrics in consideration lead to a good routing scheme. A cost is associated with each link of the network. This cost (ϕ_a) is a convex piecewise linear function of the link load. The objective function they try to minimise is the sum over all links of this cost ($\phi = \sum_{a \in A} \phi_a$). We will later refer to this objective function as *Fortz*. We have noticed that this function, though empirical, could be seen as a linear approximation of $\frac{l_a}{1-u_a}$. At low link utilisation, $1 - u_a \approx 1$ and $Fortz \approx min \sum_{a \in A} l_a$, while at high utilisations, $\frac{1}{1-u_a}$ becomes significant, leading to a load balancing policy (avoiding links with high utilisation). There is no OD pair consideration in this objective function. Many papers have reused this objective function.

3.2 MIRA

In [2], Kodialam et al. introduce the concept of minimum interference routing. They propose an objective function which is a weighted sum of the maxflows over all possible source-destination pairs on the residual topology. Their online algorithm, called MIRA, is a heuristic that tries to maximise this objective function. Formally, the objective function to be maximised is $\sum_{(s,t)} \alpha_{st} \theta_{st}$, where

[4] The amount of traffic that can be routed on the residual network is in fact the sum over all links of the available bandwidth. Indeed one obvious (and degenerated) solution to the max throughput problem is to associate traffic only with the pairs of nodes that are located at the extremities of a link. We can associate with these pairs the available bandwidth on the corresponding link.

α_{st} is a weight associated with the ingress-egress pair (s, t). The weights associated with ingress-egress pairs are administrative weights that determine the relative importance of the ingress-egress pairs to the network administrator. Behind this objective function, the goal is to minimise the blocking probability of a future new request, without information about it. The idea is that if the maxflow between one source and one destination decreases, this means that the maximum request that can be accepted between these two nodes decreases as well. Thus, the *MIRA* objective function is characterising likely good future. There is no embedded metric characterising good current state. We will see later in the simulations the implications of this fact.

3.3 Blanchy

In [3], Blanchy et al. present an online heuristic traffic engineering algorithm to optimise a load balancing objective function. The pure load balancing objective function is $\sum_{a \in A}(u_a - u_{mean})^2$ with $u_{mean} = \frac{1}{|A|}\sum_{a \in A} u_a$, the mean link utilisation in the network. *This function is the variance on the link utilisation and, as such, represents the deviation from the optimal load balancing situation.* To limit the length of the paths of a pure load balancing function, they add a "shortest path" term and arrive at the following objective function: $\sum_{a \in A}(u_a - u_{mean})^2 + \alpha \sum_{a \in A}(u_a)^2$. It is interesting because *the (weighted) combination of both terms will give more importance to the load-balancing term if the deviation is high enough to justify the detour, else it will let the "shortest path" term minimise the resources used. The weighted factor α allows to give more importance to one aspect or to the other.* This objective function does not directly include TE metrics we introduced in section 2.1. It does not include a delay contribution and there is no consideration about OD pairs. The traffic minimisation term tries to minimise the size of the paths.

3.4 Delay

In [4], Elwalid et al. associate a cost with each link. They try to minimise the total cost which is the sum over all links of the link cost. The cost of a link is a function of the link load. They assume that this function is convex. They say that a natural choice for the link cost is the delay so that their network-wide cost function is defined as $MeanDelay = \sum_{a \in A} \frac{1}{c_a - l_a}$. In section 2.1 we called this function the (unweighted) mean link delay, if we do not take the propagation delay into account. We introduce a new delay objective function (referred to as $WMeanDelay$) which is $\sum_{a \in A} \frac{l_a}{c_a - l_a}$, the weighted mean delay. Note that this objective function can also be formulated using only u_a as $WMeanDelay = \sum_{a \in A} \frac{u_a}{1 - u_a}$. These objective functions are metrics characterising good current state.

3.5 Degrande

In [5], Degrande et al. propose to maximise an objective function which is the sum of four terms: F(airness), T(hroughput), B(alance) and (network) U(tilisation).

A coefficient (named C_F, C_T, C_B or C_U) is associated to each term to give more influence to one or another. Fairness and Throughput are traffic oriented objectives while Balance and Utilisation are resource oriented objectives. Balance is defined as: $B = 1 - u_{max}$. Network utilisation is defined as $U = \sum_{a \in A} u_a$. We will not consider Fairness and Throughput in our formulation because it is not possible to express these in our LP formulation. The balance is a metric characterising likely good future. The utilisation term will minimise the size of the paths. There is no OD pair consideration and no delay contribution in this objective function. Some papers only try to minimise the maximum link utilisation. This is equivalent to *Degrande* objective function where $C_B = 1$ and $C_U = 0$. We will refer to this objective function as u_{max}.

Degrande objective function where $C_B = 0$ and $C_U = 1$ is a function which minimise $U = \sum_{a \in A} u_a$. This objective is also minimised by a classical SPF routing considering link weights equal to the inverse of their capacities. In fact, inverse capacity routing (recommended by CISCO) gives the optimal value of U. We will thus refer to this objective function as *InvCap*. We prove this by contradiction. If it is not the case, this means that there exists one flow for which the *InvCap* path does not minimise its contribution to $\sum_{a \in A} u_a$. But its contribution to this sum is in fact the traffic on this flow multiplied by the sum of the inverse of the capacity of all the links of the path, which is minimised by *InvCap* SPF.

3.6 Summary

Clearly, all the presented objective functions are related, while quite different. A first difference is that some of them use only absolute values of the load l (like $MIRA$), some only relative values u (like *Blanchy*, $WMeanDelay$, or *Degrande*) and finally some use both (like $Fortz$, or $MeanDelay$).

Table 2 presents a summary of all the presented objective functions. *MIRA*'s function is used with $\alpha_{st} = 1, \forall(s,t)$. For *Blanchy*, we have to fix the α parameter. For *Degrande*, we have to fix C'_B and C'_U. In the table, we have added the cost function called *MinHop*. This function simply minimises the total load over all

Table 2. Summary of objective functions

	Score Function (to be minimised)
Fortz	$\sum_{a \in A} \phi_a$
MIRA	$-\sum_{(s,t)} \theta_{st}$
Blanchy	$\sum_{a \in A}(u_a - u_{mean})^2 + \alpha \sum_{a \in A}(u_a)^2$
MeanDelay	$\sum_{a \in A} \frac{1}{c_a - l_a}$
WMeanDelay	$\sum_{a \in A} \frac{l_a}{c_a - l_a}$
InvCap	$\sum_{a \in A} u_a$
u_{max}	u_{max}
Degrande	$C_B.u_{max} + C_U.\sum_{a \in A} u_a$
MinHop	$\sum_{a \in A} l_a$

the links of the networks ($\sum_{a \in A} l_a$). Following the same development as for *InvCap*, this function is minimised by a SPF routing considering a weight of 1 for each link (what we call a min hop routing).

We can point out that at low load, $1 - u \approx 1$ and $c - l \approx c$ and thus $Fortz \approx MinHop$ while $WMeanDelay \approx InvCap$.

4 Simulations

In order to compare all the objective functions, we will model the traffic engineering routing problem as a linear program (LP) and solve it to optimality for all the presented objective functions. In this formulation, all the flows can be arbitrarily split. Obviously, this cannot be really implemented in a network, but can be approached with MPLS routing and to some extent with ECMP. This assumption allows us to formulate the problem as a linear program (which is easy to solve to optimality) instead of a mixed integer program (which cannot be solved to optimality in a reasonable time). The LP formulation will be used to solve the routing problem to optimality and compare the solutions obtained for every objective function. We will not write the formulation of all the objective functions, because this would take too much space, but we explain clearly how they can be reproduced. We have used an LP node-link formulation (as in [1]). $Fortz$ is expressed in [1]. For $MIRA$, we use a classical *max flow* formulation for each pair of nodes. For *Blanchy*, the square function is approximated by its linear approximation in the range $[-1, 1]$. $MeanDelay$ and $WMeanDelay$ are approximated by convex piecewise linear functions. *Degrande* and $MinHop$ are linear so they can be expressed easily, without modification. We will not present $MeanDelay$ in our result tables because we have noticed similar results than $WMeanDelay$ (noted *Delay* in the tables). For *Degrande* function, we use $C'_B = 10^3$ and $C'_U = 1$ (as in [5]) and for *Blanchy*, $\alpha = 3$ (which seems to provide good results).

4.1 Simulation Description

We made our simulations on three different networks. The first topology was generated in the TOTEM toolbox [6] using Waxman's method [7]. This topology is composed of 25 nodes and 50 full-duplex links. We set the value for parameters α and β to 0.15 and 0.2. We have generated a random traffic matrix for this topology. The second topology is an operational network. This operational network is composed of about 20 routers and 40 bidirectional links. To build a realistic traffic matrix, we have collected netflow data on each interface of the network and aggregated this information to build a traffic matrix (the procedure to generate traffic matrices from netflow traces is described in [8]). The last topology is the US research network (Abilene). It is composed of 11 nodes and 14 bidirectionnal links of 10 Gbps each. As for the operational network, we have used netflow data measured on the network to build a realistic traffic matrix. We have run our simulation on two traffic matrices per topology: the actual one (TM) and the double of it (2TM) (where each OD component has been multiplied by 2).

4.2 Results

We have to keep in mind that we made some linear approximation of some objective functions. The original function could give slightly different results in some cases. Also, some (non-linear) objective functions give different results depending on the load of the links. So, the particular traffic matrices and networks on which we made the tests can have its influence as well. Notice that $InvCap$ and $MinHop$ objective functions do not provide exactly the same routing scheme than classical shortest path first algorithm with inverse capacity or unitary metrics. Indeed, our LP model of these objective functions allows extensive and non-equal flow splitting (which is not the case in classical OSPF or ISIS implementations). So our results may present better solutions than the ones obtained by shortest path first algorithms. We have also noticed a negative point for some objective functions: multiple routing schemes achieve the optimal objective function value (especially for u_{max}). By default, the LP solver returns one of these solutions, at random. As we did not want random values in our tables, we have added a small delay contribution to these objective functions ($MIRA$, $InvCap$, u_{max}, $Degrande$ and $MinHop$) so that the LP solver returns a "good" solution from the set of possible equivalent routing schemes. So we should keep in mind that these objective functions could lead to worse results than the ones presented in this section if we do not add this delay contribution and if we do not allow arbitrary flow splitting.

Table 3. Results on network of 25 nodes (Waxman topology). The table contains absolute optimal values (in bold, green, without parentheses), or relative non-optimal values (between parentheses) with respect to the optimal one. The values that are less than 10% from the optimal value are bold. Finally the values that are 2 times worse than the optimal one are in italic and red. For each metric, we present the values for the actual traffic matrix (TM) and for the doubled traffic matrix (2TM).

Objective function	u_{max}		u_{per10}		u_{mean}		ABW_{min}		l_{mean}		θ_{tot}	
	%		%		%		Mbps		Mbps		Mbps	
	TM	2TM	TM	2TM	TM	2TM	TM	2TM	TM	2TM	TM	2TM
$Fortz$	(1.14)	(1.14)	(1.28)	(1.33)	(1.26)	(1.21)	(0.67)	*(0.55)*	**(1.03)**	**(1.05)**	**(0.97)**	**(0.95)**
$MIRA$	*100*	*100*	(1.40)	(1.48)	(1.17)	(1.16)	*0.0*	*0.0*	(1.15)	(1.10)	**6504**	**5012**
$Blanchy$	(1.22)	(1.23)	**26.0**	**50.0**	(1.13)	(1.12)	(0.88)	**531**	(1.12)	(1.11)	**(0.96)**	**(0.94)**
$Delay$	(1.20)	**(1.08)**	(1.17)	(1.20)	**(1.04)**	(1.11)	**882.0**	**(0.95)**	(1.16)	(1.11)	**(0.97)**	**(0.95)**
$InvCap$	*(2.07)*	*100*	(1.55)	(1.61)	**15.7**	**31.5**	**882.0**	*0.0*	(1.21)	(1.20)	**(0.98)**	**(0.96)**
u_{max}	**34.9**	**69.7**	(1.15)	(1.20)	**(1.07)**	(1.12)	(0.74)	*(0.57)*	(1.17)	(1.11)	**(0.97)**	**(0.95)**
$Degrande$	**34.9**	**69.7**	(1.35)	(1.39)	**(1.05)**	**(1.05)**	(0.74)	*(0.57)*	(1.19)	(1.18)	**(0.97)**	**(0.95)**
$MinHop$	*100*	*100*	(1.29)	(1.43)	(1.27)	(1.25)	*0.0*	*0.0*	**781**	**1578**	**(0.97)**	**(0.95)**

Tables 3 and 4 give the values of the TE metrics at the optimum for each objective function on Waxman and the operational networks. We do not present results on Abilene network due to lack of space. We have removed the θ_{min} metric from the tables because all the objective functions obtained the optimal value. We have also removed the $delay_{mean}$ metric because corresponding values were very small, or infinite (when $ABW_{min} = 0$). Indeed, we do not take the propagation delay into account and the link capacities are huge. This implies that all the delay values are almost equivalent, because negligible when compared to the propagation delays. Although all the delay values (except infinite values, of course) are tiny, we can point out that the *Delay* objective function gives good results for all the TE metrics on all the topologies. This is because the delay objective embeds most TE concerns (load, utilisation, available bandwidth) and even though the queuing delays are most often negligible, they become non-linearly sufficiently high when the load approaches the capacity to enforce load balancing.

We start our analysis with table 3, which presents results for the topology generated using Waxman's model. We can see that all the objective functions are not equivalent. $MinHop$ is given for comparison purposes (it gives the lowest achievable value for l_{mean}) but is clearly not a good objective function on its own. Indeed, it leads to a high value of u_{max} which is a very important concern. The lowest achievable value for u_{max} is given by the u_{max} function which only optimises this variable. The lowest achievable value of the u_{mean} variable is given by $InvCap$. This function is not very good on its own because it leads in this case to a high u_{max} value. The combined *Degrande* is a very good objective function on this topology. Indeed, it gives nearly optimal values for all the metrics. *Blanchy*, $Fortz$ and *Delay* are quite good. We notice also that $MIRA$ is good except for u_{max} which is 100% (and thus $ABW_{min} = 0$ and $delay_{mean}$ is infinite). We analyse this fact as follows. $MIRA$ is based on max-flows (and only on max-flows). Suppose that we have two routes in the network for a particular OD pair of nodes. The value of the residual max-flow will be the same if we route all the traffic on one route or if we route half of it on each route. This is the cause of the bad load balancing policy and the high value of u_{max} given by $MIRA$.

To better discriminate the *Degrande*, *Delay* and *Blanchy* functions we we can analyse the results corresponding to the doubled traffic matrix. In this case,

Table 4. Results on the operational network. See the legend of table 3 to understand these values.

Objective function	u_{max} %		u_{per10} %		u_{mean} %		ABW_{min} Mbps		l_{mean} Mbps		θ_{tot} Gbps	
	TM	2TM	TM	2TM	TM	2TM	TM	2TM	TM	2TM	TM	2TM
$Fortz$	(1.18)	(1.13)	(1.63)	(1.17)	(1.17)	(1.14)	(0.89)	(0.56)	**(1.00)**	**(1.04)**	**(0.99)**	**(0.98)**
$MIRA$	(1.41)	*100*	(1.63)	(1.65)	**(1.07)**	**(1.09)**	(0.75)	*0.0*	**(1.03)**	**(1.05)**	**4331**	**4027**
Blanchy	(1.16)	(1.15)	**14.2**	**28.5**	**(1.07)**	(1.11)	**(0.90)**	*(0.50)*	(1.24)	(1.23)	**(0.99)**	**(0.97)**
Delay	**(1.04)**	**(1.02)**	(1.32)	(1.21)	**(1.01)**	**(1.02)**	**(0.97)**	**(0.92)**	(1.15)	(1.17)	**(0.99)**	**(0.99)**
$InvCap$	(1.18)	**(1.09)**	(1.51)	(1.49)	**6.9**	**13.8**	(0.89)	(0.69)	(1.19)	(1.19)	**(0.99)**	**(0.98)**
u_{max}	**38.4**	**76.9**	(1.56)	(1.21)	**6.9**	**(1.01)**	**95.7**	**36.0**	(1.20)	(1.15)	**(0.99)**	**(0.99)**
Degrande	**38.4**	**76.9**	(1.51)	(1.49)	**6.9**	**13.8**	**95.7**	**36.0**	(1.19)	(1.19)	**(0.99)**	**(0.98)**
$MinHop$	(1.36)	*100*	(1.76)	(1.60)	(1.16)	(1.20)	(0.78)	*0.0*	**262**	**525**	**(0.99)**	**(0.98)**

Blanchy has a quite high value of u_{max}, while *InvCap* leads to a fully loaded link ($u_{max} = 100\%$). *Degrande* and *Delay* are in this case the best objective functions. $Fortz$ is also quite good in this situation.

On table 4 we can see the results for the operational topology. *Blanchy* obtains good values for all the metrics and the best value of u_{per10}. $MIRA$ logically gives the optimum for the θ_{tot} variable, which is its objective function. We remark that many other objective functions give values close to this optimal θ_{tot} value. On the operational network, we consider that the best compromise is *Degrande* because it gives almost optimal values for all the variables except u_{per10}. Both *Delay* and *Blanchy* are quite good and give better results for u_{per10}. $Fortz$ improves l_{mean} at the expense of all the other variables. $MIRA$ and $MinHop$ give high values regarding u_{max}.

We have noticed on the Abilene network that there is less variation between the values of our metrics. But we have still pointed out the performance of *Delay* and *Degrande* which are the best objective functions of these simulations.

One last important point is the fact that at low load, we can see that $Fortz$ is approaching the optimal value of l_{mean}, the objective of $MinHop$, while *Delay* is approaching the optimal value of u_{mean}, the objective of *InvCap*. This confirms the approximation we made in section 3.6.

Table 5. Metrics At Low Load (LL) and High Load (HL)

Objective	u_{max}		u_{per10}		u_{mean}		ABW_{min}		l_{mean}		θ_{tot}	
Function	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
$Fortz$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	±	±	$\surd$	$\surd$	$\surd$	$\surd$
$MIRA$	•	•	•	•	$\surd$	$\surd$	•	•	$\surd$	$\surd$	$\surd\surd$	$\surd\surd$
Blanchy	$\surd$	•	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	•	$\surd$	$\surd$	$\surd$	$\surd$
Delay	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
InvCap	•	•	$\surd$	±	$\surd\surd$	$\surd\surd$	±	•	$\surd$	$\surd$	$\surd$	$\surd$
Degrande	$\surd\surd$	$\surd\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

To conclude this section, we analyse table 5 which presents the good ($\surd$) and bad (•) metrics for each objective function at low and high load[5]. On this table, we see that $Fortz$, *Delay* and *Degrande* are the best because these have no red point.

5 Conclusion

In this paper, we have shown how well-known network-wide objective functions reflect requirements for Traffic Engineering. As our results reflect, they are not equivalent. We have shown the power of some functions and the weaknesses of others. We have outlined that, although the transmission + queueing delay is

[5] In this table, $\surd\surd$ is used to denote the optimal value and ± to denote a value which is not bad, but which is not as good as $\surd$ values.

often negligible, choosing this delay as objective function gives good results for almost all TE metrics. It is not that surprising considering that almost all TE link metrics feed into the delay (see figure 1).

The best objective functions are *Delay* and *Degrande* on the tested topologies. We have a preference for *Delay* because it does not need any configuration or parameter. *Fortz* is quite good also in all the situations, while having performance somewhat under *Delay* and *Degrande*. *Blanchy* has good results also, except for highly loaded networks. *MIRA* gives good solutions concerning the total residual max flow, but this function gives bad results concerning the maximal link utilisation.

This study provides an objective basis to select an objective function when designing a new Traffic Engineering routing algorithm. It may also be useful to revisit existing TE algorithms to make them work with the objective functions that best match the various TE concerns we have studied. Furthermore, while this study has been performed for packet switched networks, the objective functions and TE metrics used (see table 5) are also valid in circuit switched networks.

Acknowledgments

This work has been partially supported by the Walloon Region (TOTEM project) and the European Union under the E-NEXT project FP6-506869.

References

1. B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proc. of IEEE INFOCOM*, pages 519–528, 2000.
2. M. S. Kodialam and T. V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. In *Proc. of IEEE INFOCOM*, pages 884–893, 2000.
3. F. Blanchy, L. Mélon, and G. Leduc. An efficient decentralized on-line traffic engineering algorithm for MPLS networks. *Proc. of 18th ITC*, pages 451–460, 2003.
4. A. Elwalid, C. Jin, S. H. Low, and Indra Widjaja. MATE: MPLS adaptive traffic engineering. In *Proc. of IEEE INFOCOM*, pages 1300–1309, 2001.
5. N. Degrande, G. Van Hoey, P. de La Vallée-Poussin, and S. Van den Busch. Inter-area traffic engineering in a differentiated services network. *J. Networks Syst. Manage.*, 11(4), 2003.
6. G. Leduc, H. Abrahamsson, S. Balon, S. Bessler, M. D'Arienzo, O. Delcourt, J. Domingo-Pascual, S. Cerav-Erbas, I. Gojmerac, X. Masip, A. Pescaph, B. Quoitin, S.F. Romano, E. Salvatori, F. Skivée, H.T. Tran, S. Uhlig, and H. Ümit. An Open Source Traffic Engineering Toolbox. To appear in Computer Communications, 2006.
7. B.M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1671–1622, Dec 1988.
8. S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon. Providing public intradomain traffic matrices to the research community. *SIGCOMM Comput. Commun. Rev.*, 36(1):83–86, 2006.

Variable Step Fluid Simulation for Communication Network

Hongjoong Kim[1,*] and Junsoo Lee[2]

[1]Korea University, Seoul, Korea
hongjoong@korea.ac.kr
[2] Sookmyung Women's University, Seoul, Korea
jslee@sookmyung.ac.kr

Abstract. We propose a variable step fluid model for communication network in this paper. Our main goal in this research is simulation speedup of a packet-level simulator while maintaining the accuracy. The variable step fluid model not only reduces complexity but also accurately estimates simulation details such as round trip time, queue sizes, TCP windows, and packet drops. In addition, the variable step fluid model reduces event explosions, ripple effects, which have been observed in the traditional fluid models. We validate our model against `ns-2` simulation with a mixture of TCP and UDP flows under various background traffic scenarios. Our model achieves significant speedup compared to packet-level simulators. For example, the speedup of our fluid model for 20 Mb bottleneck is 40 to 70 against `ns-2`.

1 Introduction

Packet-level simulators have been widely used for performance evaluation of communication network because of their accuracy. However, packet-level simulators do not scale to large network or high bandwidth because they track every event in the system. Simulation of network traffic in a packet-level simulator such as `ns-2`[1] considers arrivals and departures of each packet at all routers and queues between a source and a destination. As the topology of the network becomes complicated and as the size of network connections increases, packet-level simulators show significant performance degrade.

Many methods have been proposed to speed up packet-level simulators. One method to overcome such an event explosion in a packet-level simulation is to use fluid models [2, 3, 4, 5, 6, 7], which abstract discrete packets with a continuous fluid flow. These fluid models average out small variations in packet-level assuming that a network traffic can be considered as a continuous flow rather than discrete packets.

While traditional fluid models [3, 5] reduce complexity of packet-level simulators, they have several drawbacks. Since packet events are discrete in nature,

[*] This research is supported by the MIC, under the ITRC support program supervised by the IITA.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 87–97, 2006.

a fluid model may lose accuracy against packet model, especially if we compare short term behaviors. Secondly, when several flows go through a network connection link, a traditional fluid model [3] often increases complexity because of the event explosion called "ripple effect". A ripple effect has been introduced in [3]. Liu et. al in [3] assume flow rates are held constant between some events, but if these events occur too frequently, performance of a fluid model is worse than that of packet models due to event explosions.

In this paper, we propose a variable step fluid model for faster simulation of communication networks. The variable step fluid model provides accurate estimation of interest measures such as size of TCP congestion window, queue sizes, the round-trip time, and the throughput. Also, our fluid model reduces the ripple effect significantly by allowing variable step size integration.

A method to reduce ripple effects with fixed step size integration has been introduced in [8], but our approach improves accuracy and performance further by varying the step size. We conduct simulation experiments on TCP congestion control to validate our model. `ns-2` packet-level simulator is chosen to validate our model. In these experiments we observe our model achieves significant speedup while maintaining little error against `ns-2`. For example, the speedup of our fluid model for 20 Mb bottleneck is 40 to 70 against the `ns-2`.

The rest of the paper is organized as follows. In Sect. 2, related work is presented. In Sect. 3, we introduce the variable step fluid model algorithm. We show experimental results in Sect. 4, and conclude in Sect. 5.

2 Related Works

There has been much research on the network simulation based on fluid models [3, 4, 9, 8]. Liu et al. [3] compare packet-level simulations and fluid models in terms of relative efficiency. They have observed ripple effects in detail and shown that fluid models perform worse than packet models when ripple effects start to occur. However, the accuracy of fluid models has not been fully analyzed.

[4] also studies trade off between packet-level and fluid models. The change of event rates in fluid and packet-level simulations with respect to the number of nodes or the number of flows are compared. However, they do not consider the accuracy of fluid models against packet models.

A scalable model of a network of AQM routers is presented in [9], and the transient behavior of the average queue length, packet loss probabilities, and average end-to-end latencies have been observed. It is shown that their fluid model is accurate and requires substantially less time to solve, especially when workloads and bandwidths are high. It is also shown that the computational complexity grows linearly with the size of the network, whereas the growth of the complexity for discrete event simulators is super-linear. However, this model is based on the expected values of variables and the accuracy of results only applies when there exist large number of flows. A fluid model introduced in [9] also captures average window and queue sizes, but these statistics are not very accurate compared to the statistics of packet-level simulations.

[8] introduces a time-driven fluid simulation model for high speed networks. Time is partitioned into *fixed-length* intervals. [8] observes only single class fluid and it does not simulate background flows. The propagation delays between any two nodes are assumed to be zero or multiples of the discretization interval length in [8], while the delay in reality depends on the traffic conditions. [8] concerns only how much traffic is generated by each source, not the exact event arrival time. In addition, computations are performed on a very simplified scenario and backlogged fluid depends on arrived fluid only.

3 Algorithm of Variable Step Size Fluid Model

3.1 Motivation

Although fixed constant time step [8] may reduce the ripple effect, simulation time and accuracy can be improved further if we use variable time step fluid model. For example, suppose that the round-trip time is τ ms and the congestion window size of TCP is α. Then α packets are emitted for τ ms. Assume that TCP sends α packets within $\tau/2$ ms and no packet is sent out for the remaining $\tau/2$ ms as in Fig. 1. If the fixed time step is smaller than the round-trip time, for example, $\tau/2$ ms, the TCP flow rate is $2\alpha/\tau$ during the first half of RTT and 0 during the second half of RTT. This introduces unnecessary rate changes. On the other hand, if constant time step is much bigger than the round-trip time, for example, multiples of τ ms, it may not accurately capture interest measures such as congestion window size of TCP because TCP varies congestion window size every RTT (τ ms).

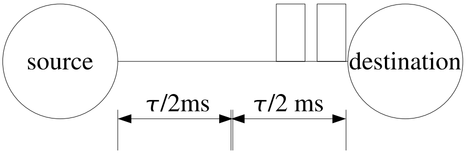


Fig. 1. Timestep

The variable step fluid model discretizes time using the round-trip time (RTT) as an interval. Since the RTT varies with the traffic condition, the moments when network variables change their values cannot be captured by a fixed time-step model if fixed-step is bigger than RTT. For example, when the network is congested (i.e. RTT is large), integration step can be large in the variable step size model. On the other hand, when the network is not congested (i.e. RTT is short), integration step need to be short to accurately capture TCP window size. The discretization in variable step fluid model also enables us to reduce the ripple effect. Ripple effects occur if flow rate is computed whenever the rate changes. For example, flows in [3] trigger an event, even though the input amount is smaller than the size of a single packet. Since our fluid model computes the flux and other variables every round-trip time, an event explosion such as ripple effect due to frequent rate change does not occur.

3.2 Algorithm

This section describes the algorithm of our variable step size fluid model. We begin with a simple case with a router, one TCP and one UDP flow. The algorithm can be easily extended to multiple routers with many TCP and UDP flows. Our fluid model algorithm is summarized in Fig. 2.

The algorithm first defines the network topology, background and foreground flows. For example, let $b(t)$ denote the available bandwidth at time t. Then $b(t)$ is initialized to (bandwidth × RTT). Then the algorithm replaces packets during RTT as a continuous flow. Our fluid model computes UDP traffic and

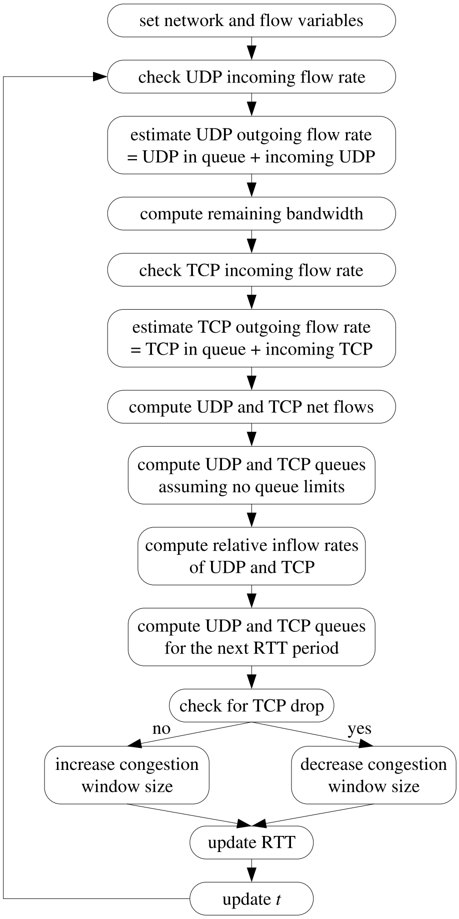

set network and flow variables
check UDP incoming flow rate
estimate UDP outgoing flow rate = UDP in queue + incoming UDP
compute remaining bandwidth
check TCP incoming flow rate
estimate TCP outgoing flow rate = TCP in queue + incoming TCP
compute UDP and TCP net flows
compute UDP and TCP queues assuming no queue limits
compute relative inflow rates of UDP and TCP
compute UDP and TCP queues for the next RTT period
check for TCP drop
no
yes
increase congestion window size
decrease congestion window size
update RTT
update t

Fig. 2. Algorithm

allocates network resources for UDP first because they do not reduce the sending rates even when packets are lost in the middle of transfer. We vary portion of background UDP flows throughput the simulation. Thus, the first step is to compute followings for UDP flows. Let $u_U(t)$ and $v_U(t)$ denote the amount of inflow and outflow of UDP flow at t, respectively. Then

$$\begin{aligned} u_U(t) &= (\text{inflow rate}) \times \text{RTT} \\ v_U(t) &= \min\{q_U(t) + u_U(t), b(t)\} \\ b(t) &= b(t) - v_U(t) \end{aligned}$$

where $q_U(t)$ is the amount of UDP flow in the queue at t. Then the remaining resources will be shared by TCP flows. Let $u_T(t)$ and $v_T(t)$ denote the amount of inflow and outflow of TCP flow at t, respectively. Then,

$$\begin{aligned} u_T(t) &= \mathit{cwnd}(t) \times (\text{packet size}) \\ v_T(t) &= \min\{q_T(t) + u_T(t), b(t)\} \\ b(t) &= b(t) - v_T(t) \end{aligned}$$

where $\mathit{cwnd}(t)$ is the congestion window size for TCP at t and $q_T(t)$ is the amount of TCP flow in the queue at t. The next step computes *net flows* between inflows and outflows for UDP and TCP flows:

$$n_i(t) = u_i(t) - v_i(t), \quad i = U, T$$

Let q_U^∞ and q_T^∞ be queue sizes when it is *temporarily* assumed there are no queue limits. Then,

$$q_i^\infty(t) = q_i(t) + n_i(t), \quad i = U, T$$

Let $q^\infty(t) = q_U^\infty(t) + q_T^\infty(t)$. Let us denote relative inflow rates of UDP and TCP by $r_i(t) = u_i(t)/u(t), \quad i = U, T$, where $u(t) = u_U(t) + u_T(t)$. When there are not sufficient resources for TCP, TCP flows will compete for limited resources and reduce sending rates accordingly if flow loss occurs. Flow loss (for UDP or TCP) occurs when the queue exceeds its maximum size, denoted by $q_{\max}$. The increase or decrease of the queue depends on net flows and there are four cases:

Case I. ($n_U > 0$ and $n_T > 0$)

If $q^\infty(t) > q_{\max}$, define

$$q_i^0 = q_i(t) + (q_{\max} - q(t))r_i(t), \quad i = U, T$$

where $q(t)$ is the queue size at t. Otherwise, define

$$q_i^0(t) = q_i^\infty(t), i = U, T$$

q_U^0 and q_T^0 are queue lengths for UDP and TCP flows at (t + *new* RTT). Since *new* RTT is not known yet, q_U^0 and q_T^0 are used temporarily now and stored to $q_U(t+\mathit{new}\ \text{RTT})$ and $q_T(t+\mathit{new}\ \text{RTT})$ when *new* RTT is obtained.

Case II. ($n_U < 0$ and $n_T < 0$)
Define

$$q_i^0(t) = \max\{0, q_i(t) + n_i(t)\}, \quad i = U, T$$

Case III. ($n_U > 0$ and $n_T < 0$)
Define $q_T^0(t) = \max\{0, q_T^\infty(t)\}$. If $q_U^\infty(t) + q_T^0(t) > q_{\max}$, define

$$q_U^0(t) = q_{\max} - q_T^0(t)$$

Otherwise, define

$$q_U^0(t) = q_U^\infty(t)$$

Case IV. ($n_U < 0$ and $n_T > 0$)
Define $q_U^0(t) = \max\{0, q_U^\infty(t)\}$. If $q_T^\infty(t) + q_U^0(t) > q_{\max}$, define

$$q_T^0(t) = q_{\max} - q_U^0(t)$$

Otherwise, define

$$q_T^0(t) = q_T^\infty(t)$$

Then, RTT is updated with the sum of the transmission delay, the queueing delay and the propagation delay. If TCP flow loss occurs in Case I or IV, the inflow rate is reduced by halving $cwnd(t)$ and threshold, $h(t)$. When TCP flow loss does not occur, $cwnd(t)$ and $h(t)$ increase exponentially up to the receiver window in the slow-start phase and linearly after that. If the window size reaches the receiver window, the slow-start phase changes to the congestion avoidance phase. After setting $q_U(t+\text{RTT}) = q_U^0(t)$, $q_T(t+\text{RTT}) = q_T^0(t)$ and $q(t+\text{RTT}) = q_U(t+\text{RTT}) + q_T(t+\text{RTT})$, t can be updated to $t+\text{RTT}$. Note also that there is no trade-off of adjusting the step size every RTT because RTT itself is the step size.

4 Simulation Experiments

In order to validate our model, we use the parking-lot topology in Fig. 3. This topology has multiple bottleneck links, one between nodes R2 and R4, and another between nodes R6 and R8. Foreground flows are sent from R1 to R10. Then, the first set of background traffic is sent from R3 to R5. The second set of background traffic is sent from R7 to R9. A similar type of topology was also considered in [9].

We consider several network scenarios by assigning different bandwidths from 5 Mb to 20 Mb to these bottlenecks. Drop Tail Queues (FIFO) are used for each router. Different types of queues such as AQM can be easily considered and are postponed for future work. We simulated extensive scenarios on this network topology, but we will show two example scenarios in this paper. Mainly,

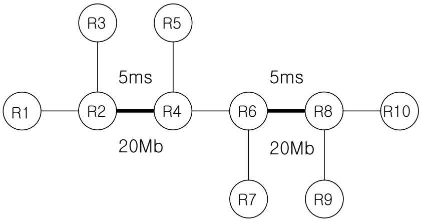


Fig. 3. Parking Lot Topology

we simulate three types of flows on this topology, which are foreground TCP, foreground UDP and background UDP flows. TCP flows go from node R1 to node R10 via nodes R2, R4, R6 and R8. Foreground UDP flows use the same path as TCP flows. Background UDP flows propagate from node R3 to node R5 passing through nodes R2 and R4, and from node R7 to node R9 passing through nodes R6 and R8. Background UDP flows represent background traffic on the network which is composed of short lived TCP flows, whereas foreground UDP assumes UDP packets are generated by protocols such as RTP[10].

4.1 TCP Flows Without Background Traffic

We first consider a flow generated by a single TCP connection without any background traffic. TCP starts at 0.5 second and is observed for 20 seconds. Once a TCP connection is established, a source sends a file of size 50 MB to the destination. We run the same scenario on `ns-2` to compare results between our model against packet-level simulations. For packet simulations in `ns-2`, the packet size is set to 1500 bytes, and the queue may hold up to 50 packets. The initial threshold of the congestion window is set to infinite. Fig. 4 shows that RTT, congestion window size, queue size, and throughput of our variable step size model captures those statistics of `ns-2` with little error when the bottleneck bandwidth is 20 Mb. As RTT increases, the congestion window size increases exponentially and the queue becomes filled up with the surplus from the inflow. When the queue becomes full, flow loss occurs and this leads to the halving of the congestion window size. Correct estimation of RTT change in the variable step fluid model allows us to detect the change of the congestion window size and the queue precisely as shown in Fig. 4. The throughput of our model also captures the result of `ns-2`.

4.2 Foreground TCP and UDP Flows with Background UDP Flows

Now suppose that there are 3 foreground TCP, 20 foreground UDP and 50 background UDP flows. The simulation is performed for 50 seconds, and three TCP connections share the same bottleneck. Each TCP connection is initiated at 0.5 second and transfers a file of size 50 MB. In this scenario 10 foreground UDP flows are sent from R3 to R5 and R7 to R9 at 10 seconds. Each UDP flow sends out 240 Kb of data per second. At 20 seconds in simulation time,

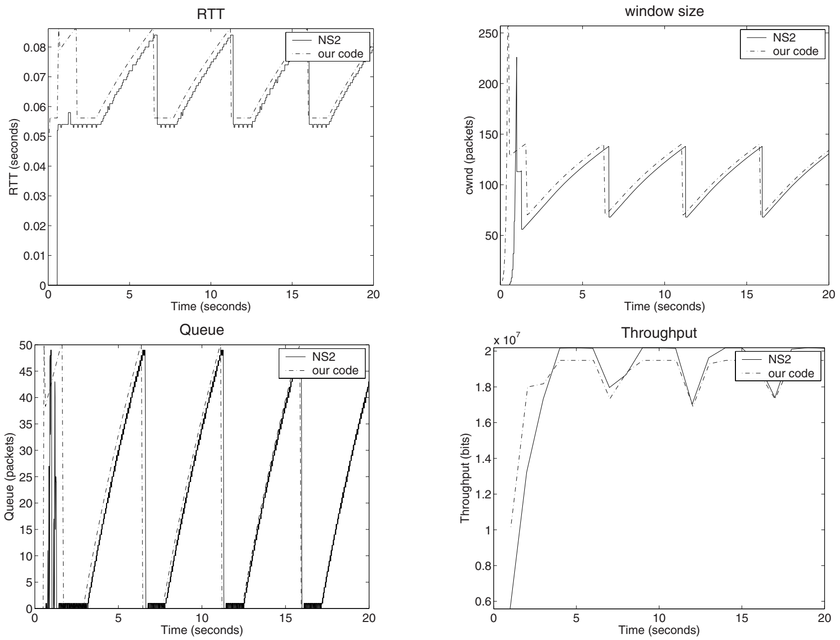


Fig. 4. Round-trip time (top left), congestion window size (top right), Queue size (bottom left) and throughput (bottom right) from 1 TCP flow when bottleneck bandwidth is 20 Mb

10 more foreground UDP flows are sent from R3 to R5 and R7 to R9 with 240Kb each. In 30 seconds, the entire foreground UDP flows are disconnected. Since each UDP source sends 12Kb of packet for every 0.05 second, UDP flows occupy about 2.4 Mb of the bottleneck bandwidth for 10 seconds, and 4.8 Mb of the bottleneck bandwidth for the next 10 seconds. In addition, background UDP traffic is injected for the last 10 seconds. The sending rate of background UDP traffic is 48 Kb and its on and off time is 500 ms, respectively. Thus each background On-Off source generates 24 Kb per second on average.

A traditional fluid model may consider fluid chunks using packets averaged out during on-time. Our fluid model introduces larger scale to define a single UDP flow. Since a packet is of size 12 Kb, the number of packets sent by each UDP flow follows the exponential distribution with mean 2. Thus 50 background UDP flows generate about 1.2 Mb per second on average in `ns-2`. Similarly, the background UDP flow in our fluid model sends out a flow at a constant rate of 1.2 Mb every second from its source to the destination. TCP flows merge with one UDP flow between nodes R2 and R4 and merge with another UDP flow from node R6 through node R8 to node R9.

Fig. 5 shows the results from `ns-2` and the fluid model when bottleneck bandwidth is 20 Mb. Round-trip time, congestion window size, and throughput are all captured with little error. The adjustment of the congestion window size due to dynamic injection of UDP flows are matched closely between `ns-2` and the

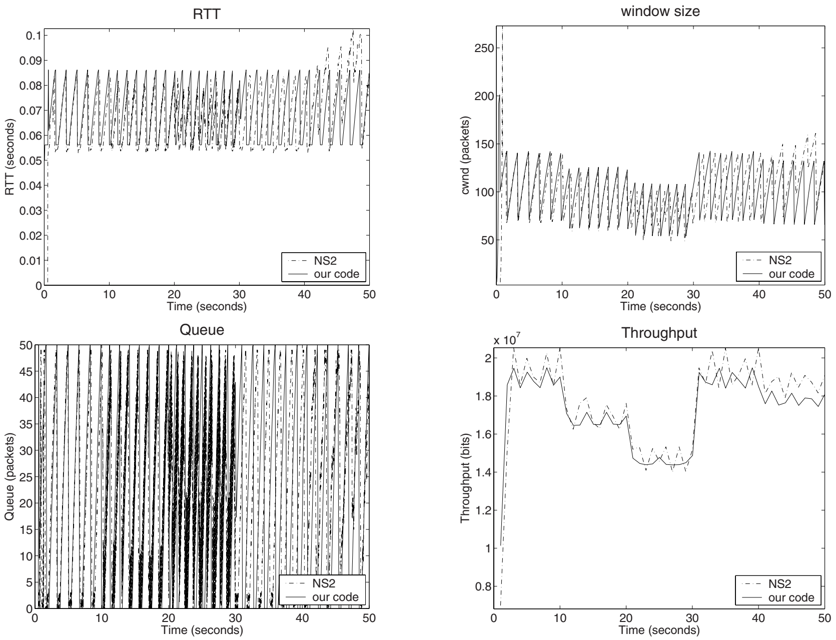


Fig. 5. RTT (top left), congestion window size (top right), Queue size (bottom left) and throughput (bottom right) from 3 TCP, 20 UDP and 50 background UDP flows when bottleneck bandwidth is 20 Mb

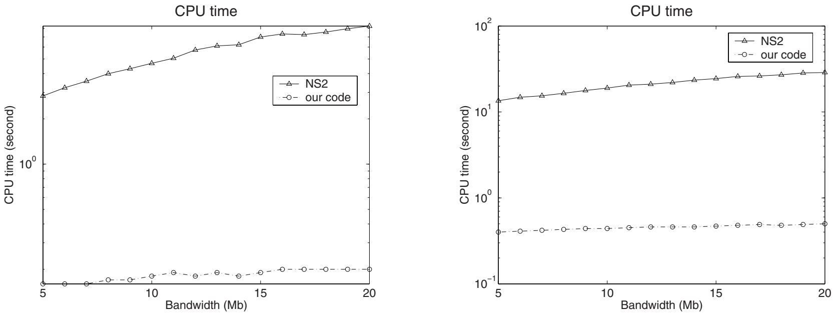


Fig. 6. CPU time for 1 TCP flow(left), CPU time for 3 TCP, 20 UDP and 50 background UDP flows(right)

fluid model. Fig. 5 also shows that our model captures RTT and throughput computed from `ns-2`.

Now we consider the speedup of variable step fluid model compared to `ns-2` packet-level simulation. Left of Fig. 6 compares computation times between `ns-2` and our fluid model simulations, the case study in Sect. 4.1, when the bandwidth of bottlenecks changes. Note that the CPU time is plotted in log-scale. This shows that our fluid model reduces the computational cost significantly. In this comparison, the fluid model takes about 0.2 second to run. `ns-2`, on the other hand, takes about 2.8 up to 8.4 second depending on the bandwidth. When the bottleneck bandwidth is 20 Mb, Running time in `ns-2` is 42 times more than that of the fluid model. As the bandwidth increases, more packets are generated in the packet level simulator, thus `ns-2` would require more computation time. Right of Fig. 6 compares the CPU time in the second scenario in Sect. 4.2, where 3 TCP, 20 UDP and 50 background UDP flows exist. In this scenario, the CPU time for the fluid model is about 0.4 second, whereas `ns-2` takes 13.3 to 28.2 seconds depending on the bandwidths of the bottleneck link. When the bottleneck bandwidth is 20 Mb, the speed up of fluid model over `ns-2` is around 70. Thus, our variable step fluid model achieves significant speedup if we compare CPU time against that of packet-level simulations.

5 Conclusion

In this paper, a variable step fluid model has been introduced to simulate network traffic in the communication network. Our model replaces discrete packet-level events with a propagation of continuous fluid flow. The variable step fluid model estimates the variation of queues accurately and captures the round-trip time, the congestion window size, and the throughput of TCP connections. With the compensation of negligible error, the variable step fluid model reduces the computational load. To validate our fluid model against `ns-2`, two network traffic scenarios are considered. When single TCP flow is simulated, we showed that our fluid model saves up to 97.5% of CPU time compared to `ns-2` packet-level simulator. If there are 3 TCP flows, 20 foreground UDP flows, and 50 background UDP flows, we save up to 99% of CPU time. We consider general TCP and UDP model in this paper. Modeling of different TCP implementations such as Reno, NewReno, SACK, etc or modeling of flows other than TCP or UDP such as TERC flows can be implemented into the current fluid model as a traffic source module. The current fluid model assumes identical pair of a source and a destination for each TCP flow. When there are multiple pairs of TCP flows, the computation time increases linearly and it is reserved for future research.

References

1. The VINT Project, a collaboratoin between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC: The ns Manual (formerly ns Notes and Documentation). (2000) Available at *http://www.isi.edu/nsnam/ns/ns-documentation.html.*
2. Ahn, J.S., Danzig, P.B.: Packet network simulation: speedup and accuracy versus timing granularity. IEEE/ACM Trans. on Networking **4**(5) (1996) 743–757
3. Liu, B., Figueiredo, D.R., Yang Guo, J.K., Towsley, D.: A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation. In: Proc. of the IEEE INFOCOM. Volume 3. (2001) 1244–1253

4. Liu, B., Guo, Y., Kurose, J., Towsley, D., Gong, W.: Fluid simulation of large scale networks: Issues and tradeoffs. In: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications. Volume IV. (1999) 2136–2142
5. Guo, Y., Gong, W., Towsley, D.: Time-stepped hybrid simulation (TSHS) for large scale networks. In: Proc. of the IEEE INFOCOM. (2000)
6. Kumaran, K., Mitra, D.: Performance and fluid simulations of a novel shared buffer management system. In: Proc. of the IEEE INFOCOM. (1998) 1449–1461
7. Liu, B., Figueiredo, D.R., Yang Guo, J.K., Towsley, D.: A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation. In: Proc. of the IEEE INFOCOM. Volume 3. (2001) 1244–1253
8. Yan, A., Gong, W.B.: Time-driven fluid simulation for high-speed networks. IEEE Transactions on Information Theory **45**(5) (1999) 1588–1599
9. Liu, Y., Presti, F.L., Misra, V., Towsley, D., gu, Y.: Fluid models and solutions for large-scale ip networks. In: ACM SIGMETRICS. (2003)
10. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard) (2003)

Estimating Link Capacity in High Speed Networks*

Ling-Jyh Chen[1], Tony Sun[2], Li Lao[2], Guang Yang[2],
M.Y. Sanadidi[2], and Mario Gerla[2]

[1] Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan
[2] Department of Computer Science, UCLA, Los Angeles, CA 90095, USA

Abstract. Knowledge of bottleneck capacity of an Internet path is critical for efficient network design, management, and usage. With emerging high speed Internet links, most traditional estimation techniques are limited in providing fast and accurate capacity estimations. In this paper, we propose a new technique, called PBProbe, to estimate high speed links. PBProbe is based on CapProbe; however, instead of solely relying on packet pairs, PBProbe employs a "packet bulk" technique and adapts the bulk length in order to overcome the well known problem with packet pair based approaches, namely the lack of accurate timer resolution. As a result, PBProbe not only preserves the simplicity and speed of CapProbe, but it also correctly estimates link capacities within a much larger range. Using analysis, we evaluate PBProbe with various bulk lengths and network configurations. We then perform emulation and Internet experiments to verify the accuracy and speed of PBProbe on high speed links. The results show that PBProbe is consistently fast and accurate in the great majority of test cases.

1 Introduction

Estimating the bottleneck capacity of an Internet path is a fundamental research problem in computer networking; knowledge of such capacity is critical for efficient network design, management and usage. In the past few years, with the growing popularity of emerging technologies such as overlay, peer-to-peer (P2P), sensor, grid and mobile networks, it is becoming increasingly desirable to have a simple, fast and accurate tool for capacity estimation and monitoring. To accommodate the diversity in network arrangements, an ideal capacity estimation tool must also be scalable and applicable to a variety of network configurations.

A number of techniques have been proposed for capacity estimation on generic Internet paths [1, 2, 3, 4, 5, 6, 7]. Among them, CapProbe [5] and Pathrate [3] have been well accepted as two fast and accurate tools in generic network scenarios. However, CapProbe is a round-trip estimation scheme that works well only on paths consisting of a symmetric bottleneck link. Pathrate, on the other hand, is based on histograms and may converge slowly when the initial dispersion measurements are not of unimodal. As a result, CapProbe has difficulty estimating capacities of asymmetric links [8], and Pathrate performs poorly on wireless links [5].

* This work is co-sponsored by the National Science Council and the National Science Foundation under grant numbers NSC-94-2218-E-001-002 and CNS-0435515.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 98–109, 2006.

To address the problems above, specialized capacity estimation tools have been proposed for specific and emerging network scenarios. For instance, ALBP [9] and AsymProbe [8] are intended for capacity estimation on asymmetric links, and AdHoc Probe [10] aims to estimate the end-to-end path capacity in wireless networks. However, for emerging high speed network links (i.e., gigabit links), recent studies have showed the standing challenging to estimate high speed link capacity (various system issues) [11], and a simple, fast and accurate technique is still lacking and desirable.

In this paper, we propose a capacity estimation tool for high speed network links, called PBProbe. PBProbe is inspired by CapProbe. However, instead of solely relying on one pair of packets, PBProbe employs the concept of "Packet Bulk" to adapt the number of probing packets in each sample in accordance to the dispersion measurement. More specifically, when the bottleneck link capacity is expected to be low, PBProbe uses one pair of packets as usual (i.e. the bulk length is 1). For paths with high bottleneck capacities, PBProbe increases the bulk length and sends several packets together, which enlarges the dispersion between the first and and last packet, to overcome the known timer resolution problem. As we will discuss in more detail later in the paper, timer resolution is the main challenge in estimating high capacity link capacities [11, 12].

The rest of the paper is organized as follows. In section 2, we summarize related work on capacity estimation. In section 3, we present and describe PBProbe. In section 4, we present an analysis of PBProbe and evaluate the speed and accuracy of PBProbe with Poisson cross traffic. In section 5, we evaluate PBProbe on high speed links in our emulator testbed as well as on the Internet. Section 6 concludes the paper.

2 Related Work

Previous research on capacity estimation relied either on delay variations among probe packets as illustrated in Pathchar [4], or on dispersion among probe packets as described in Nettimer [6] and Pathrate [3]. Pathchar-like tools (such as pchar [2] and clink [1]) have limitations in accuracy and speed as shown in [5] [13]. Moreover, they evaluate the capacity of a link based on the estimates of previous links along the path, thus estimation errors accumulate and amplify with each measured link [7].

Dispersion-based techniques suffer from other problems. In particular, Dovrolis' analysis in [3] showed that the dispersion distribution can be multi-modal due to cross traffic, and that the strongest mode of such distribution may correspond to either (1) the capacity of the path, or (2) a "compressed" dispersion, resulting in capacity overestimation, or (3) the Average Dispersion Rate (ADR), which is always lower than capacity. Another dispersion-based tool, SProbe [7], exploits SYN and RST packets of the TCP protocol to estimate the downstream link capacity, and employs two heuristics to filter out samples which have experienced cross traffic. However, SProbe does not work properly when the network is highly utilized [14].

Unlike the above approaches, CapProbe [5] uses both dispersion measurements and end-to-end delay measurements to filter out the packet pair samples that were distorted by cross traffic. This method has been shown to be both fast and accurate in a variety of scenarios. The original implementation of CapProbe uses ICMP packets as probing packets, and it measures the bottleneck capacity on a round-trip basis. As a result, the capacity estimate does not reflect the higher capacity link when the path is asymmetric. Other difficulties are encountered when intermediate nodes employ priority schemes to delay ICMP packet forwarding (e.g. Solaris operating system limits the rate of ICMP responses, and it is thus likely to perturb CapProbe measurements) [15].

Recent capacity estimation studies have extended the target network scenarios to more diverse environments. For instance, Lakshminarayanan et al. have evaluated estimation tools of capacity and available bandwidth in the emerging broadband access networks [16]. Chen et al. extended CapProbe to estimate *effective path capacity* in ad hoc wireless networks [10]. In addition, ABLP [9] and AsymProbe [8] have been proposed for capacity estimation on the increasingly popular asymmetric links (e.g. DSL and satellite links).

Nonetheless, capacity estimation on high speed links remains a challenge. Though recent studies have verified the accuracy of Pathrate in estimating gigabit links [17], however, the evaluation was done on an emulator-based testbed, which cannot represent realistic Internet dynamics. Thus, an experimental evaluation of capacity estimation on high speed links is still lacking.

In this paper, we propose a novel packet bulk technique for estimating high speed link capacities, called PBProbe. PBProbe is based on CapProbe, but it probes the bottleneck link capacity using UDP packets (instead of the ICMP packets used by CapProbe). We present the PBProbe algorithm in the next section.

3 Proposed Approach: PBProbe

In this section we introduce PBProbe. Similar to CapProbe, PBProbe estimates the link capacity by actively sending a number of probes to the network and using the *minimum delay sum* filter to identify the "good" sample. However, instead of employing a packet pair, PBProbe uses *packet bulk* of length k in each probing and measures the capacity for each direction separately. Specifically, there are two phases in PBProbe. In the first phase, PBProbe estimates the capacity of the *forward* link; whereas in the second phase, PBProbe estimates the capacity of the *backward* link. Fig. 1 illustrates the algorithm of PBProbe.

In the first phase (as shown in Fig. 1-a), host A first sends a *START* packet to host B to initiate the estimation process. Once the process is initiated, B sends a *Request To Send (RTS)* packet to A every G time units. Upon the receipt of the *RTS* packet, host A immediately sends B a packet bulk of length k (note: bulk length = k means that $k + 1$ packets are sent back to back). For the i-th probing sample, suppose B sends the *RTS* packet at time $t_{send}(i)$ and receives the j-th packet (in the i-th sample) at time $t_{rcv}(i, j)$. The delay sum (i.e. S_i) and the dispersion (i.e. D_i) of the i-th packet bulk sample are given by:

$$S_i = (t_{rcv}(i, 1) - t_{send}(i)) + (t_{rcv}(i, k + 1) - t_{send}(i)) \tag{1}$$

$$D_i = t_{rcv}(i, k + 1) - t_{rcv}(i, 1) \tag{2}$$

If none of the $k + 1$ probing packets experience cross-traffic induced queueing, the sample will reflect the correct capacity. Thus, the "good" sample (say, the m-th sample)

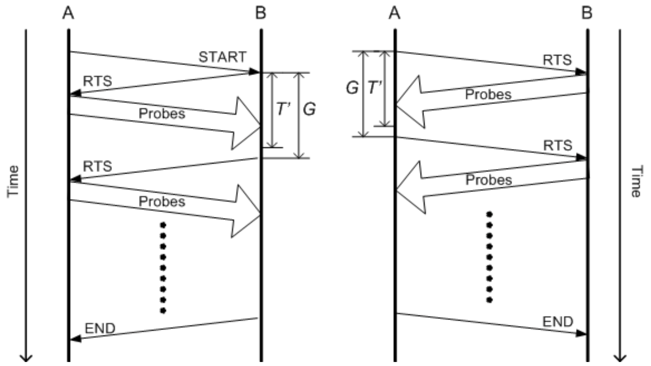


Fig. 1. Illustration of PBProbe (a) Phase I: measuring forward direction link capacity; (b) Phase II: measuring backward direction link capacity

is identified by applying the minimum delay sum filter to all probing samples (say, n samples):

$$m = arg \min_{i=1\ldots n} S_i \tag{3}$$

Therefore, the capacity estimate is made by using the dispersion of the m-th sample with the minimum delay sum:

$$C = \frac{kP}{D_m} \tag{4}$$

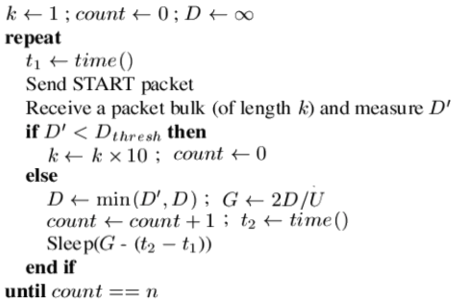
where P denotes the packet size of each probing packet. Since the packet bulk samples are delivered only in the forward direction, the estimated capacity corresponds to the bottleneck in this direction.

Once the first phase ends, B sends an *END* packet to A, and PBProbe enters the second phase (as shown in Fig. 1-b). In this phase, A first sends an *RTS* packet (every G time units) to B, and B replies a packet bulk of length k right upon the receipt of each *RTS* packet. Similar to the first phase, PBProbe measures the delay sum and dispersion for each sample, and estimates the capacity in the backward direction by using the minimum delay sum filter.

3.1 The Inter-sample Period: *G*

PBProbe probes the link capacity by sending a packet bulk every G time units. The value of G is critical for the convergence time of PBProbe. The larger G is, the slower PBProbe estimation becomes. However, G can not be too small either. If it is too small, PBProbe is more likely to create congestion in the networks and thus requires longer time to converge. Therefore, for PBProbe, we set G to be:

$$G = \frac{2D_{m'}}{U} \tag{5}$$



```
$k \leftarrow 1$ ; $count \leftarrow 0$ ; $D \leftarrow \infty$
repeat
  $t_1 \leftarrow time()$
  Send START packet
  Receive a packet bulk (of length $k$) and measure $D'$
  if $D' < D_{thresh}$ then
    $k \leftarrow k \times 10$ ; $count \leftarrow 0$
  else
    $D \leftarrow \min(D', D)$ ; $G \leftarrow 2D/U$
    $count \leftarrow count + 1$ ; $t_2 \leftarrow time()$
    Sleep($G$ - ($t_2 - t_1$))
  end if
until $count == n$
```

Fig. 2. The algorithm for determining the appropriate bulk length, k, in PBProbe

where $D_{m'}$ is the dispersion of the good sample (i.e., $D_{m'} = \frac{kP}{C}$), which has the minimum delay sum among all probing samples seen so far, and U is the maximum network utilization allowed for PBProbe estimation. We provide a short proof below showing that *if* $G = \frac{2D_{m'}}{U}$, *the network utilization constraint, U, can be guaranteed.*

Proof. Since $k \geq 1$, we know that

$$G = \frac{2D_{m'}}{U} \geq \frac{D_{m'}}{U} + \frac{D_{m'}}{kU} = \frac{kP}{CU} + \frac{P}{CU} = \frac{(k+1)P}{CU} \tag{6}$$

$$\Rightarrow \frac{(k+1)P}{G} \leq CU \tag{7}$$

Let R denote the data rate of the introduced packet bulk probes , i.e., $R = \frac{(k+1)P}{G}$; therefore we can conclude $R \leq CU$, i.e., the probing data rate is never larger than the load constraint, U.

3.2 The Packet Bulk Length: *k*

The major difference between CapProbe and PBProbe is that PBProbe sends packet bulks. The purpose of using packet bulks is to overcome the limited system timer resolution, as well as to avoid the additional latency caused by segmentation and reassembly when the packet size used is larger than the MTU. The algorithm in Fig. 2 is employed by PBProbe to automatically determine the proper bulk length, k, for capacity estimation.

In the beginning, k is initialized to 1, i.e., PBProbe behaves like CapProbe, using packet-pair to probe the link capacity. However, whenever the measured dispersion is smaller than a certain threshold, say D_{thresh}, this algorithm will increase the bulk length (k) by ten-fold and restart the estimation process. Clearly, the decision of D_{thresh} value depends on the system timer resolution. In this work, we set D_{thresh} = $1ms$ for all the experiments

3.3 The Number of Samples: *n*

CapProbe employs a sophisticated convergence test to determine whether a good sample has been obtained. To simplify the implementation, PBProbe simply estimates link capacities using a fixed number of n samples. Obviously, the larger n is, the more accurately PBProbe estimates. The required time for one PBProbe capacity estimate is linearly proportional to the value of n. More specifically, the larger n is, the longer time PBProbe requires. Based on the experimental results reported in the previous CapProbe studies [18] [5], we decide to set $n = 200$ throughout this paper.

4 Analysis

In this section, we present a queueing model that predicts the probability of obtaining a "good" sample for a single link with Poisson distributed cross traffic. For simplicity, we assume the probing samples of PBProbe arrive according to a Poisson process so that they take so to speak "a random look" at the link. We also assume that the probing samples do not constitute a significant load on the network since they are sent infrequently. Finally, we assume the buffers are large enough so that probing packets will not be dropped due to buffer overflow. The analytical model is described next, followed by the results.

Suppose the arrival and service rate of Poisson cross traffic are λ and μ, the service time of one single probing packet is τ, and the bottleneck link utilization is ρ. The probability of the first probing packet arriving to an empty system, i.e. p, is given by:

$$p = 1 - \frac{\lambda}{\mu} = 1 - \rho \tag{8}$$

Since there is no queueing delay experienced by any probing packets of a "good" sample, the probability of no queueing delay for the remaining k probing packets (i.e. no cross traffic packets arrive in the $k\tau$ period) is $e^{-\lambda(k\tau)}$. Therefore, the probability of obtaining a "good" sample, i.e. p_0, is given by:

$$p_0 = pe^{\lambda(-k\tau)} = (1-\rho)\, e^{-k\lambda\tau} \tag{9}$$

The expected number of samples, $\overline{N}$, for obtaining a good sample is then derived as:

$$\overline{N} = \sum_{n=1}^{\infty} np_0(1-p_0)^{n-1} = \frac{1}{p_0} = \frac{e^{k\lambda\tau}}{1-\rho} \tag{10}$$

Suppose the size of the probing packets and the cross traffic packets are equal, then $\tau = \frac{1}{\mu} = \frac{\rho}{\lambda}$. Therefore, $\overline{N}$ could be rewritten as:

$$\overline{N} = \frac{e^{k\rho}}{1-\rho} \tag{11}$$

The relationship between the expected number of required samples for one good sample ($\overline{N}$) and link utilization (ρ) with different packet bulk length (k) is shown in Fig. 3.

From the results illustrated in Fig. 3, when $k = 1$ (i.e. packet-pair based CapProbe), $\overline{N}$ is around 25 when the utilization (ρ) is as high as 0.9. However, as k increases, $\overline{N}$ increases exponentially. For instance, when $\rho = 0.3$[1], $\overline{N}$ is around 30 when $k = 10$,

[1] The utilization of Abilene backbone network, which is the gigabit backbone of Internet2, is hardly over 30% [19].

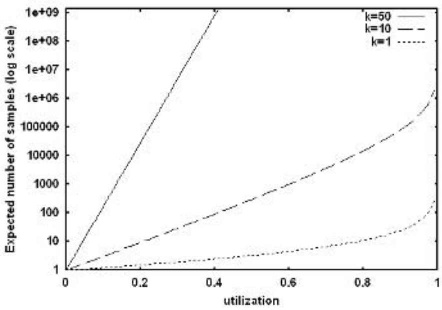


Fig. 3. The expected number of samples ($\overline{N}$) with different link utilization (ρ) and packet bulk length (k) under Poisson cross traffic

but becomes around 5,000,000 when $k = 50$. It turns out that the estimation speed of PBProbe (i.e. the required number of samples for obtaining a good sample) is highly related to the employed packet bulk length. Though employing a large packet bulk can improve the accuracy of dispersion measurements, such large bulk will need a much larger number of tries in order to lead to a good sample and therefore will slow down the estimation considerably.

5 Evaluation

5.1 Emulation Experiments

The emulator-based experiments were run on our laboratory testbed. In the scenario, three testing machines are connected serially with 1 Gbps links. The NISTNet emulator [20] is installed on the middle machine and can configure the middle machine to create bottleneck link on the gigabit path. The purpose of this set of experiments is to verify the needs of bulk length adaption for estimating high speed links; therefore, we used two fixed bulk lengths (i.e., $k = 10$ and 100) and did not employ cross traffic in the emulation experiments. We varied the bottleneck link capacity, and conducted 30 runs of the experiments for each capacity setting. The results of $k = 10$ and 100 are shown in Fig. 4.

Since no cross traffic was present in these experiments, packets within a packet bulk are expected to traverse the path back-to-back without being disturbed. Therefore, ideally, the measured dispersion of each packet bulk should represent the undistorted dispersion corresponding to bottleneck link capacity. Moreover, based on these perfect dispersions, the capacity estimate should be accurate and consistent for every probing sample.

However, since PBProbe requires accurate dispersion measurements, the capacity estimates tend to become inaccurate when timer resolution is inadequate. Table 1 shows the required timer resolution of PBProbe on links with different capacity and for different bulk lengths. Obviously, when the bottleneck link capacity is high or the bulk length is small, a high timer resolution is required. For instance, with link capacity = 100 Mbps

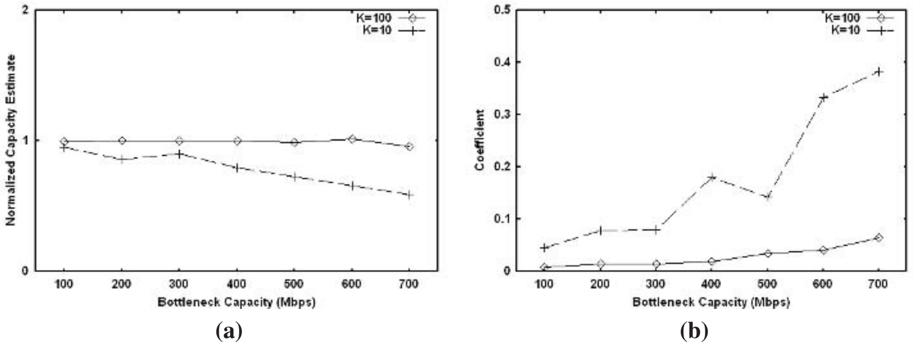


Fig. 4. PBProbe estimation results using NISTNet emulator: (a) normalized capacity estimates and (b) coefficient of variation of capacity estimates with various bottleneck capacity settings

Table 1. Required system timer resolution for accurate PBProbe estimation (assuming probing packet size is 1500 bytes)

	Bottleneck Link Capacity		
k	1Gbps	100Mbps	10Mbps
1	0.012ms	0.12ms	1.2ms
10	0.12ms	1.2ms	12ms
100	1.2ms	12ms	120ms

and packet pairs (i.e., $k = 1$), only a powerful processor can satisfy the required resolution of 0.12 ms. As the capacity increases to 1 Gbps, the required 0.012 ms resolution becomes very difficult to achieve, and the measurement accuracy will degrade. Indeed, this trend was observed in Fig. 4-a and 4-b. In these two figures, for small k, the capacity estimates become increasingly inaccurate and inconsistent when the required timer resolution became greater (or equivalently, when the bottleneck capacity was enlarged).

However, the results also reveal that, when measuring high capacity links, a large k can successfully alleviate the inaccuracy and inconsistency of capacity estimates caused by poor timer resolution. For instance, when the bottleneck link capacity is 600 Mbps, PBProbe with bulk length $k = 100$ can accurately estimate the capacity (as illustrated by very small coefficient of variation on capacity estimates). In contrast, when $k = 10$, PBProbe can only measure around 60% of the link capacity, and the coefficient of variation is much larger. Based on the experimental results as well as the analysis shown in Table 1, we conjecture that our testbed hosts can only provide timer resolution of approximately 1 ms. Therefore, we decided to fix $T_{thresh} = 1ms$ in the k adaptation algorithm of PBProbe for all the following experiments.

5.2 Internet Experiments

Here, we conducted Internet experiments to evaluate PBProbe in realistic environments. Five Internet hosts (CalTech: California Institute of Technology; GaTech:

Georgia Institute of Technology; NTNU: National Taiwan Normal University; PSC: Pittsburgh Supercomputing Center; UCLA: University of California at Los Angeles) and five Internet gigabit paths (within California: UCLA and CalTech); across country: UCLA - PSC, PSC - GaTech, GaTech - UCLA; and international: NTNU - UCLA) were selected for the experiments. The topology and path properties of the selected paths are illustrated in Fig. 5.

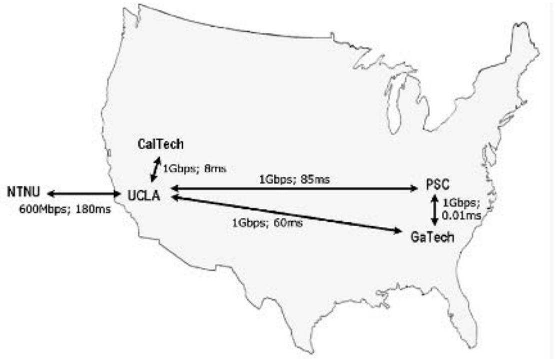


Fig. 5. Topology and path properties (bottleneck capacity and round trip delay) of selected gigabit Internet paths

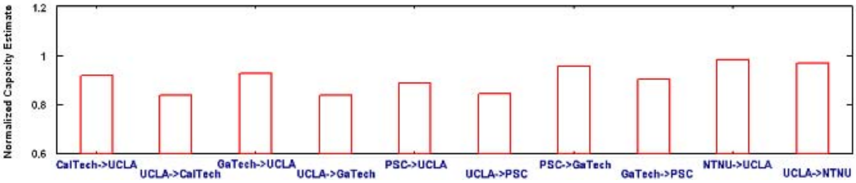


Fig. 6. PBProbe experiment results (mean of 20 runs) on high speed links

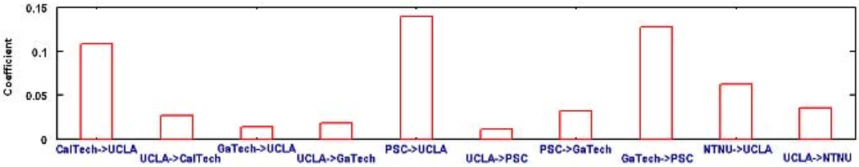


Fig. 7. PBProbe experiment results (coefficient of variation of 20 runs) on high speed links

Fig. 6 and 7 illustrate the experiment results (i.e. normalized mean and coefficient of variation of capacity estimates in 20 runs). It is clear that, in Fig. 6, the normalized capacity estimates are mostly within 90% accuracy range, except three outgoing links from UCLA to CalTech, GaTech, and PSC. This is due to the fact that the outgoing

Table 2. Comparison of PBProbe and Pathrate on Internet links. (Capacity: Mbps; Time: sec).

	PBProbe		Pathrate	
	Capacity	Time	Capacity	Time
CalTech → UCLA	919.6	14	933.2	17
UCLA → CalTech	839.4	14	945.3	1146
GaTech → UCLA	928.1	14	932.9	18
UCLA → GaTech	840.3	14	968.1	1223
PSC → GaTech	959.9	13	995.4	1122
GaTech → PSC	905.9	13	947.0	17
PSC → UCLA	889.6	14	935.6	20
UCLA → PSC	845.2	15	905.6	20
NTNU → UCLA	580.6	20	575.6	1641
UCLA → NTNU	588.4	21	573.4	1641

Table 3. Comparison of PBProbe and Pathrate overhead on Internet gigabit links

	GaTech → UCLA		UCLA → GaTech	
	PBProbe	Pathrate	PBProbe	Pathrate
spent time	14 sec	18 sec	14 sec	1223 sec
total packets	20,213	2,414	20,213	27,630
total bytes	30,319,500	3,543,752	30,319,500	39,707,740
BW consumption	2.166Mbps	1.575Mbps	2.166Mbps	0.260Mbps

link of UCLA backbone has around 30% utilization, which is much higher than the utilization of the incoming link (around 15%) [21], and, in this case, PBProbe requires a large number of samples in order to correctly estimate the capacity. Since we set $n = 200$ in all experiments, PBProbe only estimated around 80% of the link capacity.

In addition, Fig. 7 also shows that the coefficient of variation of PBProbe estimates are below 0.15 on all tested links, i.e. the capacity estimates (20 runs) of each high-speed link are very stable. Comparing with the results shown in Fig. 4, it turns out that PBProbe was able to adapt its bulk length to the most appropriate value (i.e. $k = 100$) so that it could consistently and accurately estimate the capacities of high speed links.

5.3 Comparisons of PBProbe and Pathrate

So far, we have evaluated PBProbe in a variety of network scenarios. In this subsection, we compare the performance of PBProbe and Pathrate in terms of accuracy, speed, and bandwidth consumption (i.e. overhead). The experiments are performed on the same set of high speed Internet links as illustrated in Fig. 5, and the results of capacity estimates and required time are shown in Table 2.

In Table 2, PBProbe measures at least around 85% bottleneck capacities of all tested links; whereas Pathrate is very accurate and measures at least 90% bottleneck capacities for all links. However, for some links with long delay and/or high utilization, Pathrate required more than 1000 seconds to estimate the capacity. This is due to the fact that, if the distribution of measured dispersion is not unimodal after the first phase, Pathrate will start the second phase to probe the network using different packet train lengths and packet sizes. Once Pathrate enters the second phase, it takes long time to determine the correct link capacity. As a result, Pathrate converges fast if the dispersion distribution is unimodal in the first phase, but it becomes much slower than PBProbe otherwise.

We also compared the packet overhead caused by PBProbe and Pathrate on high speed Internet links. Table 3 shows the comparison on one of the high speed links, the UCLA - GaTech link. From the experiment results, PBProbe is more expensive than Pathrate, if Pathrate only uses one phase to estimate link capacity on high speed links.

In case when Pathrate is required to enter the second phase, PBProbe and Pathrate produce a comparable amount of packet overhead. However, since Pathrate is much slower after entering the second phase, the bandwidth consumption (i.e. bits per second) is much smaller than PBProbe. It is worth pointing out that, even though the bandwidth consumption of PBProbe (approximately 2 Mbps) seems to be relatively high, it is realistically only 0.2% of the bottleneck capacity (i.e., 1 Gbps); thus, it is not intrusive to other traffic flows in the network.

The experiment results suggest that there are trade-offs between PBProbe and Pathrate for high-speed path capacity estimation. On the one hand, PBProbe yields very good estimation results rapidly (e.g., less than 20 seconds in most cases). If given more time, it will progressively improve the estimates, since better samples can be obtained. On the other hand, Pathrate tends to produce accurate results, but the required time may vary from approximately 20 seconds to 20 minutes. Therefore, Pathrate may not be ideal in scenarios when an estimation of the bottleneck capacity needs to be obtained within a very short time.

It should also be noted that the packet overhead of PBProbe is proportional to the employed bulk length k. While measuring a high speed link, PBProbe increases its bulk length and in turn increases the packet overhead, in order to overcome the limited support of system timer resolution. Nonetheless, thank to the employed U parameter, the bandwidth consumption of PBProbe is restricted by the utilization upper bound. Hence, PBProbe can carefully control the trade-off between the bandwidth consumption and the required time in order to satisfy the requirement of different applications.

6 Conclusions

In this paper, we studied a classic problem of link capacity estimation, and we proposed a technique, called PBProbe, to estimate bottleneck capacity for emerging high speed links. PBProbe is based on the CapProbe algorithm, but it uses “packet bulk” to adapt the number of packets in each probing according to different network characteristics. As a result, it preserves the simplicity, speed, and accuracy of CapProbe, as well as overcoming the poor system timer resolution problem on high speed links. Using analysis, emulation, and Internet experiments, we evaluated the accuracy, speed, and overhead of PBProbe on various network configurations. The results show that PBProbe can correctly and rapidly estimate bottleneck capacity in almost all test cases. Comparing to other capacity estimation techniques, PBProbe is ideal in real deployments that requires online and timely capacity estimation. This capacity estimation technique can further provide assistance to typical applications such as peer-to-peer streaming and file sharing, overlay network structuring, pricing and QoS enhancements, as well as network monitoring.

Acknowledgments

We are grateful to the following people for their help in carrying out PBProbe measurements: Sanjay Hegde (CalTech), Che-Chih Liu (NTNU), Cesar A. C. Marcondes (UCLA), and Anders Persson (UCLA).

References

1. "Clink: a tool for estimating internet link characteristics," http://allendowney.com/research/clink/.
2. "pchar: A tool for measuring internet path characteristics," http://www.kitchenlab.org/www/bmah/Software/pchar/.
3. C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" in *IEEE Infocom*, 2001.
4. V. Jacobson, "Pathchar: A tool to infer characteristics of internet paths," ftp://ftp.ee.lbl.gov/pathchar/. [Online]. Available: ftp://ftp.ee.lbl.gov/pathchar/
5. R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "Capprobe: A simple and accurate capacity estimation technique," in *ACM SIGCOMM*, 2004.
6. K. Lai and M. Baker, "Measuring bandwidth," in *IEEE Infocom*, 1999, pp. 235–245.
7. S. Saroiu, P. K. Gummadi, and S. D. Gribble, "Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments," in *IEEE Infocom*, 2002.
8. L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla, "End-to-end asymmetric link capacity estimation," in *IFIP Networking*, 2005.
9. Y. Lin, H. Wu, S. Cheng, W. Wang, and C. Wang, "Measuring asymmetric link bandwidths in internet using a multi-packet delay model," in *IEEE ICC*, 2003.
10. L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla, "Adhoc probe: Path capacity probing in ad hoc networks," in *WICON*, 2005.
11. G. Jin and B. Tierney, "System capability effect on algorithms for network bandwidth measurement," in *ACM IMC*, 2003.
12. R. Kapoor, L.-J. Chen, M. Y. Sanadidi, and M. Gerla, "Accuracy of link capacity estimates using passive and active approaches with capprobe," in *IEEE ISCC*, 2004.
13. K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," in *ACM SIGCOMM*, 2000.
14. S.-J. Lee, P. Sharma, S. Banerjee, S. Basu, and R. Fonseca, "Measuring bandwidth between planetlab nodes," in *PAM*, 2005.
15. S. Savage, "Sting: a tcp-based network measurement tool," in *USENIX Symposium on Internet Technologies and Systems*, 1999.
16. K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye, "Bandwidth estimation in broadband access networks," in *IMC*, 2004.
17. R. Prasad, M. Jain, and C. Dovrolis, "Evaluating pathrate and pathload with realistic cross-traffic," http://www.cc.gatech.edu/ jain/pub/talk/best03_talk.ppt, 2003 Bandwidth Estimation Workshop. [Online]. Available: http://www.cc.gatech.edu/~jain/pub/talk/best03_talk.ppt
18. L.-J. Chen, T. Sun, D. Xu, M. Y. Sanadidi, and M. Gerla, "Access link capacity monitoring with tfrc probe," in *E2EMON*, 2004.
19. "Abilene network traffic," http://loadrunner.uits.iu.edu/weathermaps/abilene/.
20. "Nistnet: network emulation package," http://www.antd.nist.gov/itg/nistnet/.
21. "Cenic network statistics," http://cricket.cenic.org/grapher.cgi.

Internet Traffic Mid-term Forecasting: A Pragmatic Approach Using Statistical Analysis Tools

Rachel Babiarz and Jean-Sebastien Bedo

France Telecom R&D Division, Innovation Economics Laboratory,
38-40 rue du Général Leclerc, 92794 Issy-les-Moulineaux Cedex 9, France
{rachel.babiarz, jeansebastien.bedo}@rd.francetelecom.com

Abstract. Network planning is usually based on long-term trends and forecasts of Internet traffic. However, between two large updates, telecommunication operators deal with resource allocation in contracts depending on the mid-term evolution of their own traffic. In this paper, we develop a methodology to forecast the fluctuations of Internet traffic in an international IP transit network. We do not work on traffic demands which can not be easily measured in a large network. Instead, we use link counts which are much simpler to obtain. If needed, the origin-destination demands are estimated *a posteriori* through traffic matrix inference techniques. We analyze link counts stemming from France Telecom IP international transit network at the two hours time scale over nineteen weeks and produce forecasts for five weeks (mid-term). Our methodology relies on Principal Component Analysis and time series modeling taking into account the strain of cycles. We show that five components represent 64% of the traffic total variance and that these components are quite stable over time. This stability allows us to develop a method that produce forecasts automatically without any model to fit.

Keywords: IP international transit network, traffic forecasting, principal component analysis, time series modeling, strain modeling.

1 Introduction

Forecasting the end to end traffic profiles of an IP network is very important in order to deal with traffic engineering tasks like new resources planning or network design. Whereas these end to end traffic demands can be directly obtained for traditional telecommunication networks based on circuit switching, it is more difficult with packet-based routing which is used in Internet. Hence, it relies on the IP protocol which does not include an accounting mechanism. The authors of [1] highlight these difficulties. Tools like Netflow from Cisco have been developed to directly measure the traffic demands, but these measurements are quite difficult to obtain in an accurate and exhaustive manner. These tools are based on sample measurements, use a lot of network resources and then cannot be activated on all routers. So, we do not have historical data on the end to

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 110–122, 2006.

end traffic demands. The only easily available historical data are the amount of traffic exchanged between adjacent routers (link counts) that can be obtained through the Simple Network Management Protocol (SNMP). The link counts correspond to the sum of several users traffic demands entering the network into one edge router and exiting at an other edge router. To obtain the traffic origins and destinations, a lot of traffic matrix inference techniques have been developed these last few years. They use link counts and routing schemes to estimate the end to end demands of the network. [2, 3, 4, 5] give a quite complete overview of the recent work done by the research community on this topic.

In this paper, our goal is then to predict the link counts. We need online forecasting to increase our reactivity and flexibility. So we propose a completely automatic technique. We test our approach on France Telecom IP international transit network. We work on several weeks of SNMP data, at a large time scale (two hours granularity) and do forecasts for a few weeks. The main difficulty is that there are usually many links (next to a thousand) in an international transit network and it is not feasible to fit a model on each link to produce its forecasts. We develop a pragmatic approach to deal with this high dimension and assure its full automation for operational purposes. Crovella et al. show in [6] that Principal Component Analysis (PCA) applied to link counts data can drastically reduce the high dimension into a few components. This is due to high correlations existing between link counts evolutions. We see that these few components stemming from PCA exhibit strong time periodicities that do not change very much. We study the cycle change of shape of the components and propose a new approach to build forecasts for this kind of data based on simple statistical tools. Our forecasting technique has the advantage to be fully automated contrary to classical time series models such as SARIMA which are needed to be fitted in several steps. The contribution of our paper to the field is to validate the PCA forecasting methodology proposed by Crovella et al. on new real traces of a large international backbone IP and the introduction of the study of the shape morphing inside the forecasting methodology itself.

The paper is organized as follows. In section 2, we briefly present previous works relative to traffic forecasting. Then, we describe the data on which we have applied our methodology in section 3. The section 4 details the way we can decompose IP traffic observed on a lot of links simultaneously in elementary shapes thanks to PCA. The forecast techniques we propose are developed in the next section. The last section is devoted to the results forecasts descriptions.

2 Related Work

Forecasting traffic was already an issue for the Plain Old Telephone Service (POTS) ([7, 8]). Most of the processing theories have been extensively used to deal with this problem. But there was no statistical multiplexing, so traffic profiles were much more continuous contrary to IP traffic profiles. Internet traffic forecasting techniques (see for example [9, 10, 11]) have mainly addressed local area network and small time scales, such as seconds or minutes, that are relevant for dynamic resource allocation and show the local effects of statistical multiplexing.

In our case, we are interested in international transit network and larger time scales which are more appropriate when doing capacity planning and network design. But long range dependencies effects begin to appear very rapidly as the time scale grows ([12]). The first work dealing with large time scale is described in [13]. In this paper, the authors predict a single value for the entire network using linear time series models which is not sufficient for network planning purposes.

The nearest work to ours is exposed in [14]. The evolution of IP backbone traffic at large time scales are modeled and long-term predicted combining wavelet multiresolution analysis and linear time series models.

3 Data

We collected SNMP data from all the routers of the France Telecom IP international transit network from April 3, 2005 to August 13, 2005. These data represent the total amount of traffic exchanged between all the adjacent routers (link counts) by ten minutes time slots. For this nineteen weeks period, about eight hundred links have been observed between next to two hundred routers. These links are either access links or core links, we do not distinguish between this two kind of links in this paper and do forecasts for both type. Among all the links, about two hundred of them are not active during all the period or have a negligible traffic amount. We do not consider these specific links for the forecasts. As we are interested in doing forecasts for network planning purposes, we average our traffic measurements across two hours intervals. We do not average our data on a larger interval because we want to keep the daily periodicities (see the next paragraph) which are an important traffic element useful for the forecasts buildings. Two hours granularity is then a good compromise.

We observe two types of traffic behavior in our link counts data. There are link counts that exhibit strong daily and weekly periodicities reflecting traditional human activities. This behavior corresponds to the largest link counts

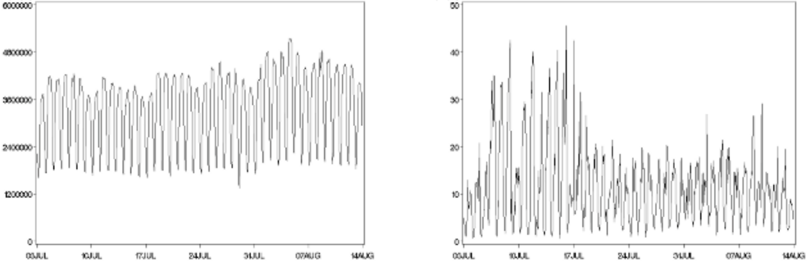


Fig. 1. Examples of a cyclic (left) and a bursty (right) link count

and represent the majority of the total traffic. Other link counts are bursty, representing occasional spikes or dips of traffic. They mainly correspond to small link counts. We show an example of these two types of traffic behavior in Figure 1. The period of time is intentionally reduced to allow to distinguish the cycles.

4 Structural Analysis of Link Counts

4.1 Principal Component Analysis Overview

In this part, we intend to introduce the Principal Component Analysis (PCA) framework but we cannot cover all aspects due to lack of space. For more details, see [6]. The method of PCA is useful to analyze a complex set of many correlated statistical variables $X = \left[X^1, ..., X^p\right]$ into new principal independent components ([15]). PCA works on zero-mean data. The principal components correspond to the eigenvectors of the covariance matrix $X^T X$:

$$X^T X \, v_i = \lambda_i \, v_i \quad i = 1, ..., p \tag{1}$$

λ_i is the eigenvalue corresponding to the eigenvector v_i. Since $X^T X$ is symmetric definite positive, its eigenvectors are orthogonal, the eigenvalues are nonnegative real and its trace, corresponding to the total variance of X, is equal to the sum of the eigenvalues, so that λ_i represents the variation part of X captured by the i^{th} eigenvector or principal component. By convention, the eigenvectors are unit vectors and the eigenvalues are sorted from large to small. Thus, the first eigenvector v_1 captures the largest variation part of X, the second eigenvector v_2 the second largest variation part, and so on. The projection of X on v_i represents the coordinate or contribution of X on the i^{th} principal component, this vector can be normalized to unit length by dividing by $\sqrt{\lambda_i}$:

$$u_i = \frac{X \, v_i}{\sqrt{\lambda_i}} \quad i = 1, ..., p \tag{2}$$

u_i is then a linear combination of the initial variables, it is usually named a score. By performing PCA, we decompose X into an optimal sum of unit rank matrices (product of a line vector by a column vector):

$$X = \sum_{i=1}^{p} \sqrt{\lambda_i} \, u_i \, v_i^T \tag{3}$$

If we consider only the first q principal components and scores, we obtain the best approximation of X with q elements. This approximation can be computed by taking p equal to q in Equation 3. In the next part, we apply the technique of PCA on our link counts data.

4.2 PCA Application on Link Counts

The matrix X, defined in the previous section, now represents the T measurements of traffic on the nineteen weeks observed period and on the L links (variables) of our network. As we have seen in section 3, the traffic data exhibit strong

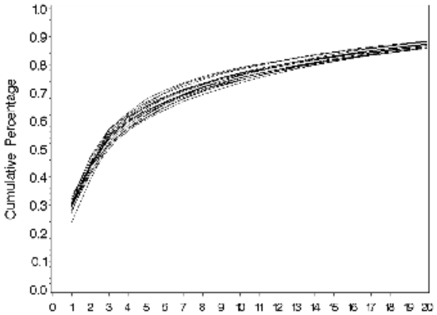


Fig. 2. Scree Plot for Link Counts

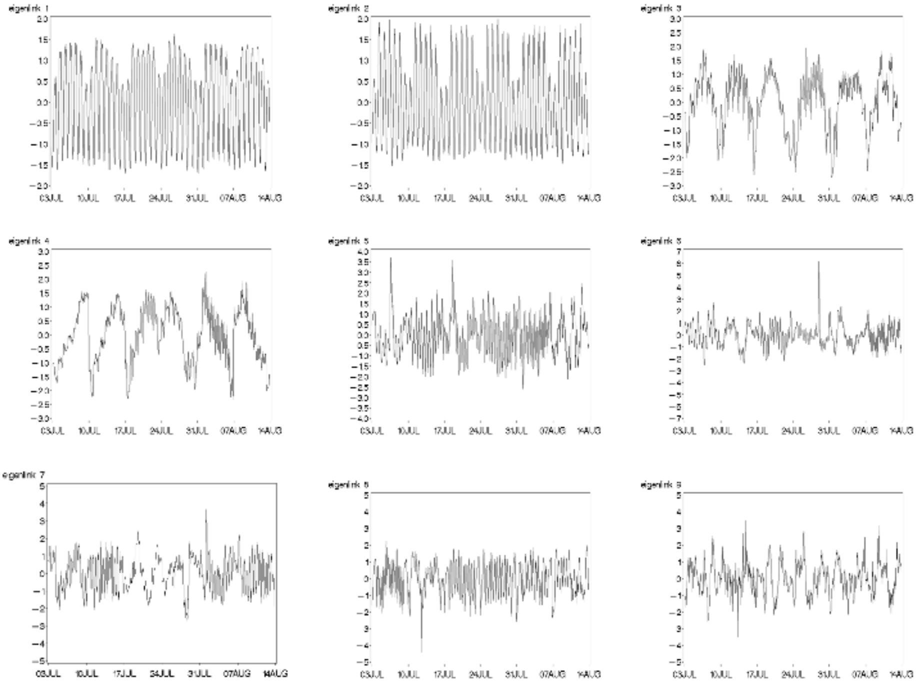


Fig. 3. First Nine Eigenlinks

daily and weekly periodicities in majority. To respect these properties, we decide to perform PCA week by week. We center and reduce the traffic evolutions for each link and per each week in order to consider traffic profile instead of raw traffic. This allows to give as much importance to all the links in the network. We obtain, for each week, L new vectors or scores, we call them the eigenlinks in reference to Crovella et al. who call them the eigenflows in [16] when PCA is applied on OD flows traffic. The eigenlinks represent the elementary shapes of the link counts. The first eigenlink captures the strongest time variation common to all links, the second eigenlink the next strongest, and so on. We represent in the figure below the variation part (cumulative percentage) captured by the first twenty eigenlinks in the form of a scree plot. A scree plot shows the sorted eigenvalues, from large to small, as a function of the eigenvalue index. Each curve corresponds to a different studied week.

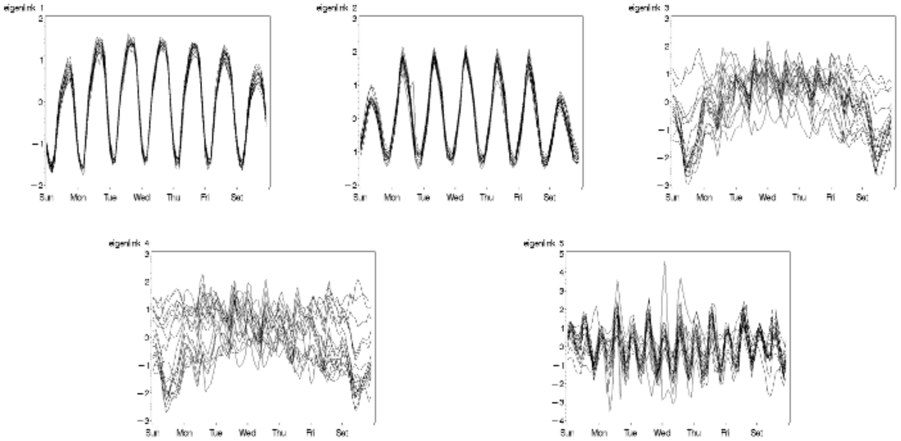


Fig. 4. First Five Eigenlinks superposed week by week

We can see that the vast majority of link counts variability is explained by the first few eigenlinks, whatever the week we consider. This is due to the fact that the majority of the link counts show the same strong daily and weekly periodicities. So, the link counts form a structure with effective dimension much lower than the total number of links.

The first nine eigenlinks are represented in Figure 3. Once again the period of time is reduced for readability reasons. The first five eigenlinks exhibit strong daily and weekly periodicities. This is not surprising as these properties concern the majority of link counts data. Their periods are different and their peaks of traffic do not appear at the same time. This is due to the different time zones covered by our network. The next eigenlinks do not reflect a particular shape, they are more bursty. In [16], a taxonomy of eigenflows is realized using heuristics based on their periodogramm and their standard deviations. The authors show that the eigenflows can be separated in three categories in a quantitative way: deterministic, spike and noise. By using the same quantitative heuristics and taxonomy, only the first five eigenlinks are characterized as deterministic. We then decide to only use the first five deterministic eigenlinks to build our forecasts for all the link counts of the France Telecom international transit network. We reduce the dimension from about six hundred to five thanks to PCA. These five elements represent 64%, in mean over the studied weeks, of the link counts total variation.

If we represent the first five eigenlinks by superposing the weeks between them (Figure 4), we can see that the link counts structure is quite stable over time. We use this property to develop our forecasting method in the next section.

5 Forecasting Techniques

We propose to build forecasts for the deterministic eigenlinks time series and then to project these forecasts on the adequate eigenvectors to obtain all the link counts forecasts via Equation 3 with q equal to the number of deterministic eigenlinks kept. It can be compared to a generalized deseasonalization method ([17]). We discuss the adequate eigenvectors choice further. As we want to develop a methodology that can be fully automated for network planners, we do not use classical linear time series models such as SARIMA models which are needed to be fitted manually one by one (see [18] for details on the Box-Jenkins methodology to fit SARIMA models). We propose a pragmatic approach based on basic statistical tools such as mean and standard deviation and using the time stability of the link counts structure stemming from the first deterministic eigenlinks we have already observed in the previous section.

The study of the ratio between a sequence mean and its standard deviation is a good way to measure the dispersion of this sequence. Indeed, the higher this ratio is, the more stable the serie is. We use this criteria to quantify the stability of each deterministic eigenlink time serie. We then build the following indicator:

$$Y_{t_1}^{t} = \frac{Mean_{\,i}\,(X_t^i - X_{t_1}^i)}{Standard\,Deviation_{\,i}\,(X_t^i - X_{t_1}^i)}\,,\,\forall\,t_1 \neq t \tag{4}$$

where t represents the sub-measurements of a cycle. In our case, the cycle corresponds to a week, so t varies from Sunday 0h-2h to Saturday 22h-0h, i.e. 84 sub-measurements in all. X_t^i represents the traffic volume for the sub-measurement t for the week i. If $Y_{t_1}^{t_2}$ is high, it means that the cycle do not change its shape between the sub-measurements t_1 and t_2. Then, if we want to forecast the value of X_{t_2} for the next cycle $K+1$ from the K observed cycles, we can do as follows:

$$X_{t_2}^{K+1} = X_{t_1}^{K} + Mean_{\,i=1}^{\,K}\,(X_{t_2}^i - X_{t_1}^i) \tag{5}$$

We generalized Equation 5 by considering all the sub-measurements t of a cycle:

$$X_t^{K+1} = \sum_{t_1 \neq t} weight_{\,t_1}^{\,t}\,[X_{t_1}^K \,+\, Mean_{\,i=1}^{\,K}\,(X_t^i - X_{t_1}^i)],\,\forall\,t \tag{6}$$

where

$$weight_{\,t_1}^{\,t} = \frac{(Y_{t_1}^t)^2}{\sum_{t_1 \neq t}(Y_{t_1}^t)^2} \tag{7}$$

Our prediction formula defined by Equation 6 consists of computing the prediction for a sub-measurement t of a cycle by a weighted sum of all the other sub-measurements. The introduction of a weight stemming from the cycle stability indicator defined by Equation 4 allows to consider more importance to the closest sub-measurements to the sub-measurement for which we compute the forecast in terms of strain between cycles. We use this technique to compute the deterministic eigenlinks forecasts. The results are given in the next section.

Once the eigenlinks forecasts are obtained, the final forecasts for all the link counts are computed by projecting these forecasts on the eigenvectors. As, we have performed PCA on our data week by week, we have a set of eigenvectors and the corresponding eigenvalues for each week or cycle. We propose to use the eigenvectors and eigenvalues stemming from the last observed cycle as we have seen that the structure is stable over time.

6 Forecasting Results

6.1 Deterministic Eigenlinks Forecasts

We divide our observation period in two parts: an estimation period from April 3, 2005 to July 9, 2005 (14 weeks) on which we develop our forecasting techniques and an evaluation period from July 10, 2005 to August 13, 2005 (5 weeks) from which we compare our forecasts results to the real data. Figure 5 shows the forecasting results we obtain for the first five eigenlinks using our methodology (Equation 6). The forecasts are in dashed lines and the real values in plain lines.

We can notice from these graphs that the first components are quite well forecasted even if their shape is smoothed by the technique. However the forecasts for the fourth component are distorted compared to real data. This is mainly due to the high strain between sequential cycles on this component.

We compare these forecasting results with those obtained when SARIMA models are fitted. The following models have been fitted: $SARIMA\,(1,0,1)_{1,12,84}$ for the first four eigenlinks and $SARIMA\,(1,0,1)_{1,6,12,84}$ for the fifth eigenlink.

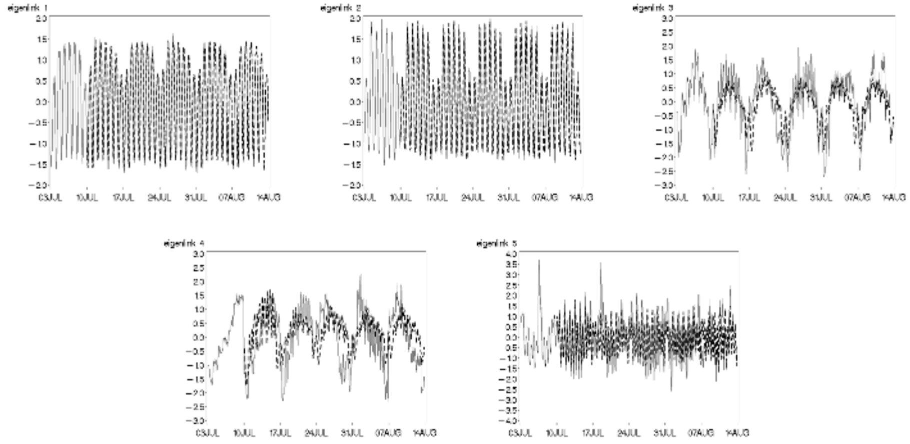


Fig. 5. First Five Eigenlinks Forecasts

For these comparisons, we compute the median of the absolute relative error (Equation 8) for each eigenlink. X corresponds to the real values and X' to the forecasts. We do not compute the mean because of some extreme values due to abnormal traffic behavior. We can see in Table 1 that our technique gives in general slightly better results than SARIMA models. We do not pretend to give better results than any SARIMA model. The SARIMA models we fit for the comparisons are choosen with the same orders for convenience (we need an automated method). In addition, the method we proposed does not rely on iterative algorithms (contrary to SARIMA models) leading to a lower computational burden.

$$Median\left(\frac{|\,X' - X\,|}{|\,X\,|}\right) \tag{8}$$

Table 1. Comparison Results (Median Error) between our method and SARIMA models

	Our Method	SARIMA
Eigenlink 1	0.1108	0.1046
Eigenlink 2	0.1324	0.1370
Eigenlink 3	0.6408	0.6787
Eigenlink 4	0.8144	0.9258
Eigenlink 5	0.5990	0.6355

6.2 Link Counts Forecasts

We remind that the forecasts for link counts are obtained thanks to Equation 3 with q=5:

$$X' = \sum_{i=1}^{5} \sqrt{\lambda_i}\, u_i\, v_i^T \tag{9}$$

λ_i and v_i correspond respectively to the i$^{\text{th}}$ eigenvalue and the i$^{\text{th}}$ eigenvector stemming from the PCA application on the last week of the estimation period, and u_i is the i$^{\text{th}}$ forecasted eigenlink over the evaluation period. Then, X' is a matrix containing the forecasts for all the link counts. As we have centered and reduced the data to zero mean and unit standard deviation before applying PCA, the final forecasts results have to be rescaled by the mean and the standard deviation of each link count respectively. We use the mean and standard deviation of link counts observed on the last week of the estimation period.

Besides the median absolute relative error defined in Equation 8, we also compute the relative errors over the median (Equation 10) and over the maximum (Equation 11), the traffic median or maximum being indicators usually used for network design.

$$\frac{Median(X') - Median(X)}{Median(X)} \tag{10}$$

$$\frac{Max(X') - Max(X)}{Max(X)} \tag{11}$$

where X corresponds to the real values and X' to the forecasts. We compute the three types of error for each link count and each week of the evaluation period. We represent in Figure 6 these errors for all the link counts classified from large to small and for the first (dashed line) and last (dotted line) week of the evaluation period.

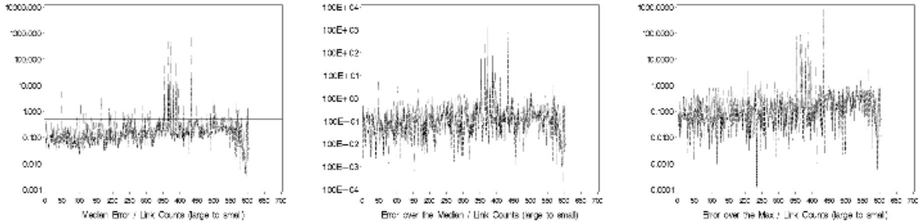


Fig. 6. Forecast Errors

We can see that the majority of link counts have a median error lying between 5% and 50% (horizontal lines). The errors have the same order of height whatever the traffic volume of the link count and are quite stable between the first and last weeks of the evaluation period. The errors peaks mainly correspond to:

- bursty link counts without cycles
- links with a traffic mean which varies a lot between weeks
- anomalies of traffic

We would not have significatively better results with other methods. This is due to the traffic sporadicity which makes this specific error peaks unpredictable. We show examples of these types of behavior in Figure 7. The forecasts are in dashed lines and the real values in plain lines.

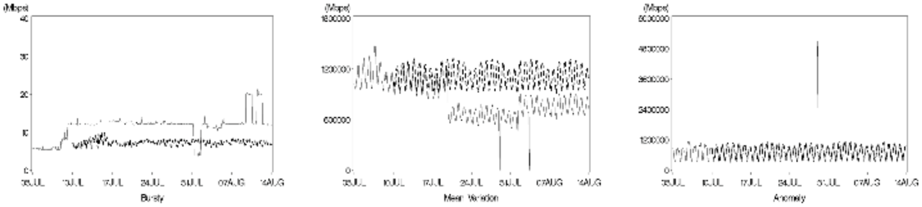


Fig. 7. Examples of Link Counts with bad Forecasts

6.3 Discussion About Traffic Matrix Estimation

In this paper, we have proposed to forecast link counts and then to obtain the end to end traffic demands thanks to traffic matrix estimation techniques based on link counts only. One could wonder how would behave our forecasting method on historical origin-destination (OD) counts directly. Some internal studies on partial traces have shown that the number of deterministic principal components are slightly the same for link counts or OD counts. In addition, the first components are very similar between links and OD pairs. As a result, traffic matrices forecasts would probably be more precise if we used our forecasting method directly on OD counts since we would not add the approximations of the traffic matrix estimation techniques. Unfortunately, measuring complete OD traffic matrices is often impossible in the case of large IP backbones. And storing historical data without interruptions during weeks concerning traffic matrices is even harder. As a consequence, it is not viable today to rely on forecastings based on direct OD counts for large networks.

7 Conclusion

In this paper, we have developed a pragmatic methodology to predict Internet traffic on all the links of an international IP transit network. Our aim is to obtain these prediction results in a fully automated way in order to be directly operational for network planners who have to deal with several traffic engineering tasks like new resources planning or network design. The low computational burden of the method even allows very reactive on demand forecasts with flexible parameters (period of interest, time scale...). Our method is intentionally simple, based on non-advanced scientific tools, again for automated reasons. In summary, this methodology involves the following steps:

- (1) Principal Component Analysis on the link counts
- (2) Determine the deterministic scores or eigenlinks which are relevant
- (3) Prediction of the deterministic eigenlinks profiles using the study of the cycle change of shape
- (4) Projection of the predicted deterministic eigenlinks on the link counts structure obtained from PCA applied in step (1)

We apply this methodology on data stemming from the France Telecom international transit network over a period of nineteen weeks, between April 3, 2005 and August 13, 2005. Five deterministic eigenlinks out of 600 are kept for summing up the link counts structure. It means a reduction of computational burden by 120. Furthermore, our strain analysis framework is faster than SARIMA techniques. The majority of link counts have a median absolute relative error lying between 5% and 50%. The largest prediction errors concern link counts with a non-constant behavior.

Therefore, through this paper, we validate on real traces from a large backbone IP network the use of the PCA technique to forecast Internet traffic profiles in a large number. We have also developed a new simple method for forecasting periodic traffic profiles based on the study of the variations of cycle shapes between successive periods. It gives us some insights to better understand the underlying trends of IP traffic. Our method can be applied on a longer period by considering the mean trend (instead of the mean of the last cycle) of the link counts data. The time stability of the link counts structure have also to be controlled, PCA which gives the link counts structure must then be reapplied from time to time.

Acknowledgements

The authors would like to thank Anne-Gaëlle Corrion, Benjamin Petiau, Jean-Luc Lutton and Vincent Martin for their helpful comments and advices on this work.

References

1. K. Papagiannaki, N. Taft, A. Lakhina. A Distributed Approach to Measure IP Traffic Matrices. In ACM IMC, October 2004.
2. A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, C. Diot. Traffic Matrix Estimation: Existing Techniques and New Directions. In ACM SIGCOM, August 2002.
3. S. Vaton, J.S. Bedo, A. Gravey. Advanced Methods for the estimation of the Origin-Destination Traffic Matrix. In Revue du $25^{\text{ème}}$ anniversaire du GERAD, 2005.
4. A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, C. Diot. Traffic Matrices: Balancing Measurements, Inference and Modeling. In ACM SIGMETRICS, June 2005.
5. Y. Zhang, M. Roughan, N. Duffield, A. Greenberg. Fast Accurate Computation of Large-scale IP Traffic Matrices from Link Loads. In ACM SIGMETRICS, June 2003.
6. A. Lakhina, M. Crovella, C. Diot. Diagnosing Network-Wide Traffic Anomalies. In ACM SIGCOM, August 2004.
7. A. Passeron, E. Etve. Modelling Seasonal Variations of Telephone Traffic. In ITC, 1983.
8. P. Chemouil, B. Garnier. An Adaptive Short-Term Traffic Forecasting Procedure using Kalman Filtering. In ITC, 1985.
9. S. Basu, A. Mukherjee, S. Klivansky. Time Series Models for Internet Traffic. In IEEE INFOCOM, March 1996.
10. C. You, K. Chandra. Time Series Models for Internet Data Traffic. In IEEE LCN, October 1999.
11. A. Sang, S. Li. A Predictability Analysis of Network Traffic. In IEEE INFOCOM, March 2000.
12. W. Leland, M. Taqqu, W. Willinger, D. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). In IEEE ACM Transactions on Networking, Vol. 2, No. 1, pp. 1-15, February 1994.
13. N. K. Groschwitz, G. C. Polyzos. A Time Series Model of Long-Term NSFNET Backbone Traffic. In ICC, May 1994.

14. K. Papagiannaki, N. Taft, Z. Zhang, C. Diot. Long-Term Forecating of Internet Backbone Traffic: Observations and Initial Models. In IEEE INFOCOM, April 2003.
15. H. Hotelling. Analysis of a complex of statistical variables into principal components. In Journal of Educational Psychology, Vol. 24, pp. 417-441, 1933.
16. A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E.D. Kolaczyk, N.Taft. Structural Analysis of Network Traffic Flows. In ACM SIGMETRICS, June 2004.
17. S. Hylleberg. Seasonality in Regression. Academic Press, Orlando, FL.
18. P. Brockwell, R. Davis. Introduction to Time Series and Forecasting. Springer, 1996.

Semantic Compression of TCP Traces

Gabriel Istrate[1], Anders Hansson[1], Sunil Thulasidasan[1],
Madhav Marathe[2], and Chris Barrett[2]

[1] CCS-5, Los Alamos National Laboratory, Los Alamos NM 87545, USA
[2] Virginia Bioinformatics Institute, Blacksburg VA 24061, USA

Abstract. We propose a new methodology, RESTORED, for model-based storage and regeneration of TCP traces. RESTORED provides significant data compression by exploiting semantics of TCP. Experiments show that RESTORED can achieve over 10,000-fold compression ratios for some really large input connections, while still being able to recover several structural and QoS measures.

1 Introduction

Traffic measurement and monitoring faces the important challenge of *data storage*; due to the large amount of information, it is simply not feasible to collect all traces. A natural approach is to keep only a fraction of the packets that traverse a given link and estimate connection characteristics from the stored packets. A number of approaches to succinctly storing network data have been proposed. For example, Cisco System's NETFLOW technology is based on the simple idea of aggregating packet information to compute flow characteristics [1], i.e., time is partitioned into slots, and a router records the total number of packets it handles in any given time slot, the total number of bytes, etc. Further, Cisco has proposed a sampled version that records information based on every Nth packet. A slightly more advanced approach is to employ sampling rate adaptation. Unfortunately, these solutions lack clear justification from a scientific standpoint: a fine time granularity is not able to provide effective data compression (since a lot of data has to be recorded), whereas too coarse time granularity leads to poor characterization of network dynamics. The latter issue is also a significant weakness of SNMP counters [2]. Other monitoring tools, such as GIGASCOPE [3], supports complex performance queries, but scalability remains a major bottleneck.

Approaches that store only a subset of traffic data face the obvious problem that the stored information might not be sufficient to capture the meaningful characteristics of TCP. For instance, while a couple of measures of quality of service (e.g. throughput) could probably be estimated by storing every Nth packet, it is likely that most measures cannot be inferred in this way. A more principled, *model-based approach* is needed.

Recent years have seen substantial advances in understanding and modeling the intricate nature of TCP traffic. Aggregate traffic can display fractal [4] and multifractal [5] characteristics at small timescales. For large enough timescales technological constraints make these correlations disappear, so that there is no long-range dependence [6]. In the presence of congestion, aggregate traffic characteristics are approximately Poisson [7]. On the other hand, individual connections have a much simpler structure;

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 123–135, 2006.

many of their aspects can be modeled by Markov chains [8]. The progress in modeling temporal aspects of TCP traces has not been matched by corresponding advances in modeling the dynamics of packet IDs. This is unfortunate, since the dynamics of packet ID can influence the overall connection dynamics: TCP is a protocol that tries to maintain packet sequence integrity, and will attempt to do so by controlling the senders' congestion window. Thus a complete understanding of TCP dynamics requires a new approach in which *packet reordering* plays a central role. As convincingly argued by Bennett, Partridge, and Shectman [9], packet reordering has many severe effects on TCP performance (see also [10]). In conclusion, receiver side models of TCP should capture both temporal and reordering aspects of TCP traces.

The main goal of this paper is to *propose a new approach called* RESTORED, REceiver-oriented STOchastic REgeneration of packet Dynamics, *for model-based storage and regeneration of TCP traces*. RESTORED uses a compression scheme based in TCP semantics, and *produces traces that are provably equivalent to the original trace with respect to a rigorously defined notion of trace equivalence*. The core functionality of RESTORED can be summarized as follows: *(i) Trace Collection:* Data is collected for each session at each destination node. *(ii) Trace Compression:* We do not require exact storage of the trace but allow lossy compression. This allows us to greatly boost the compression ratio. *(iii) Generation of Synthetic Traces from the Summary Data:* The idea is that the synthetic data retains most of the dynamic information inherent in the original packet streams. *(iv) QoS Analysis Based on the Generated Traffic:* Since the synthetic data sequences are compatible with the collected ones, we can analyze QoS measures in much the same way as we would have done on the original data.

Network Data: We use real network traces obtained by monitoring network traffic, as well as synthetic network traces generated by simulation. The real packet traces were collected during August 2001 at the border router of the Computer Science Department, University of California, Los Angeles, CA. See http://lever.cs.ucla.edu/ddos/traces for details. In particular, we have used five TCP traces, called TRACE5, TRACE7, TRACE8, TRACE9, TRACE10. We found that most of the connections in these traces are very short. For example, TRACE7 consists of 245,718 connections, but 60% of them contain only one or two packets, 80% contain at most 10, and 98% contain at most 100 packets. This is, of course, in line with the observation that a small fraction of the flows accounts for a large percentage of the total traffic [11, 12]. Estan and Varghese have argued that, for many applications, knowledge of these "heavy hitters" is sufficient [13]. Since our aim is to highlight nontrivial network dynamics, we chose to study only connections with at least 100 packets. Still, there are enough connections to allow meaningful analysis: TRACE5 contains 7582 such connections, TRACE7 contains 5662, TRACE8 contains 7936, TRACE9 contains 7612, and TRACE10 contains 3399. We also employed NS-2 to generate a synthetic TCP trace called NS-2-long, to benchmark the compression performance of RESTORED. We simulated TCP Reno with SACK and delayed ACKs; all sessions were persistent FTP data connections with a fixed payload of 1072 bytes. We simulated the classical dumbell topology with multiple sources and sinks sharing the same bottleneck link, 201 connections in total. The links connecting the sources and the sinks to the bottleneck routers had a bandwidth of 100 Mbps and a delay of 10 ms. The bottleneck link bandwidth and delay were 100 Mbps and 50 ms.

The TCP send/receive buffers were set to 64 KB, and packet drop rates were controlled by limiting the output queues of bottleneck routers to at most 50 packets. All queues were drop-tail FIFO queues. The connection start times were uniformly spread on an interval between 0 and 0.5 seconds. We ran the simulation until 842 million packets had been sent (not counting ACKs), which translates to about 4.2 million packets per connection.

2 The Macroscopic Model

Our model consists of two parts: (i) A Markovian model that captures TCP dynamics at *macroscopic* time scales. (ii) A fine-grained model that completes the macroscopic view of TCP packet reordering to *microscopic* time scales. In this section discuss we discuss the first component of the model, the Markovian model for macroscopic time scales. Let us first consider the packet IDs. Since our objective is to model packet reordering rather than data fragmentation, we make the simplifying assumption that all packets have identical payload. This allows a bijective mapping from TCP sequence numbers to packet ID numbers, with the convention that the smallest sequence number is mapped to ID 1. For simplicity, we omit discussing the slow start phase; our model can readily be extended to include it. Consider the following motivating example: a receiver may observe the following packet stream (where we only display the ID numbers of the packets, and not their arrival times)

$$\underbrace{1\ \ 2}_{\mathcal{O}}\quad \underbrace{4\ \ 5\ \ 6\ \ 3}_{\mathcal{U}}\quad \underbrace{8\ \ 9\ \ 10\ \ 7}_{\mathcal{U}}\quad \underbrace{11\ \ 12\ \ 13}_{\mathcal{O}} \tag{1}$$

The order of the IDs corresponds to the order in which packets arrive at the destination, and in our case, we see that packets 3 and 7 arrive out of order. Since TCP guarantees to deliver an ordered packet stream to the application layer, it follows that there is a need for packet buffering. One can, consequently, classify the received packets into two types: those that can be immediately passed to the application layer, and those that are temporarily buffered before delivery. In our example, packets 4, 5, and 6 are temporarily buffered, and the buffer cannot be flushed until packet 3 is received. Likewise, packets 8, 9, and 10 are temporarily buffered, and the buffer is flushed at the arrival of packet 7. A packet that marks the end of a sequence of consecutively buffered packets will be called a *pivot packet*. Packets that can be immediately delivered to the application layer are trivially pivots. In our example, packets 1, 2, 3, 7, 11, 12, and 13 are thus all pivots. This definition suggests a coarsened view of TCP with two states:

State $\mathcal{O}$: The *ordered state*, in which packets can be immediately passed to the application layer

State $\mathcal{U}$: The *unordered state*, in which there is reordering and buffering.

Each occurrence of State $\mathcal{O}$ is followed by one or more occurrences of State $\mathcal{U}$. Explicitly incorporating time, one can provide a high-leven description of a TCP connection by a sequence of triples $(s_1, p_1, t_1)\,,\ (s_2, p_2, t_2)\,,\ \ldots$, where $s_n \in \{\mathcal{O}, \mathcal{U}\}$ is the state descriptor, $p_n > 0$ is the number of packets received in state s_n, and $t_n > 0$ is the time spent in state s_n.

Coarsening TCP connections at the level of pivot packets *provides an operational motivation for assuming that the observed traffic characteristics are stationary*: after a pivot packet has been received and the buffer has been flushed from the point of view of the receiver TCP is "in the same state" in circumstances with the same congestion window and the same number of packets in transit. In contrast, some models of network traffic assume second-order stationarity without any plausible motivation—at least it is not entirely clear at what time scale this assumption is warranted [14]. In fact, for large enough time scales, traffic parameters may even exhibit nonstationarity [15].

The high-level view of network traffic will be our first component for modeling TCP sequences compatible with a given trace. More specifically, we propose a Markovian model for the previous sequence. This model is schematically illustrated in Fig. 1, and formally specified in Fig. 2. The Markovian model above yields an inference algorithm which, in turn, enables us to reconstruct synthetic TCP traces up to the coarsened level of pivot packets. Section 3 extends this definition to a complete model of TCP.

Let us now statistically validate the macroscopic model. One tool that is employed is the sample autocorrelation function, $\hat{\rho}_X(h)$, of a time-series $\{X_n\}$. This function is found by first computing the sample mean, $\hat{m}_X$, and the sample autocovariance function, $\hat{\gamma}_X(h)$, associated with $\{X_n\}$. Specifically, if we let $x_1, x_2, \ldots, x_N$ be observations of $\{X_n\}$, the sample mean is just the average, $\hat{m}_X \equiv N^{-1}\sum_{n=1}^{N} x_n$, the sample autocovariance function is the standard estimate of the autocovariance function, $\hat{\gamma}_X(h) \equiv N^{-1}\sum_{n=1}^{N-|h|}(x_{n+|h|} - \hat{m}_X)(x_n - \hat{m}_X)$, and finally, the sample autocorrelation function is obtained by normalization, $\hat{\rho}_X(h) \equiv \hat{\gamma}_X(h)/\hat{\gamma}_X(0)$. First, we have to validate Claim I, i.e. we have to assess that the sequence of states can be modeled using a Markov chain. To test this hypothesis, it is enough to show that the time-series $N_{\mathcal{U}}$, the number of consecutive occurrences of State $\mathcal{U}\}$, can be viewed as a sequence of independent samples drawn from a certain distribution (possibly different for different packet streams). We will test independence by performing the *sample autocorrelation test* [16]. This test states that approximately 95% of the sample autocorrelations of an i.i.d. sequence $x_1, x_2, \ldots, x_N$ should fall between the bounds $\pm 1.96/\sqrt{N}$ for large N (assuming finite variance observations). To test Claim I we thus formulate the following null hypothesis: observations of the time-series $N_{\mathcal{U}}$ are independently sampled from a unique underlying distribution. Based on the sample autocorrelation test, this hypothesis is then rejected with a confidence of 95% if more than 5% of the sample autocorrelation coefficients fall outside the bounds $\pm 1.96/\sqrt{N}$. We chose to analyze the five UCLA traces that were described in the introduction. For the robustness of the sample autocorrelation test we follow the recommendation provided by Box and Jenkins, who suggest that N should be at least about 50 [17]. A number of packet streams must then be discarded. Still, 236 streams are retained for TRACE5, 292 for TRACE7, 266 for TRACE8, 317 for TRACE9, and 63 for TRACE10. The percentage of packet streams that fail the sample autocorrelation test (and thus disprove the null hypothesis) is displayed in the topmost panel of Table 1.

To validate Claims II–III we need to test that (i) there is no correlation in the time-series $\{p_n, t_n\}$ associated with State $\mathcal{O}$ (or State $\mathcal{U}$), and that (ii) there is no correlation between consecutive observations of (p_n, t_n) associated with a transition from State $\mathcal{O}$ to State $\mathcal{U}$ (or from State $\mathcal{U}$ to State $\mathcal{O}$). In order to bring statistical evidence for Claim II in the definition of the Markovian model, we first consider the four time-series $P_{\mathcal{O}}$, the number of packets in State $\mathcal{O}\}$, $P_{\mathcal{U}}$, the number of packets in State $\mathcal{U}\}$, $T_{\mathcal{O}}$, the time spent in State $\mathcal{O}\}$, and $T_{\mathcal{U}}$, the time spent in State $\mathcal{U}\}$. We should test these four time series for lack of correlation. This is done by using the sample autocorrelation test, and the null hypothesis is that observations of the time series are independently sampled from four unique distributions. Once again, we analyzed the five UCLA traces, and the results are presented in the middle panel of Table 1. To complete the statistical evidence for Claim II in the definition of the Markovian model, we have to test for lack of correlation between characteristics of two different consecutive states. The test we employ is the *nonparametric Spearman test* for linear correlation between components of a bivariate time-series [18]. As null hypothesis, we assume that the two components of the four bi-variate time-series $P_{\mathcal{O}\to\mathcal{U}}$, (the number of packets in state $\mathcal{O}$, the number of packets in next state $\mathcal{U}$), $P_{\mathcal{U}\to\mathcal{O}}$, (the number of packets in state $\mathcal{U}$, the number of packets in next state $\mathcal{O}$), $T_{\mathcal{O}\to\mathcal{U}}$ (the time spent in state $\mathcal{O}$, the time spent in next state $\mathcal{U}$), $T_{\mathcal{U}\to\mathcal{O}}$, (the time spent in state $\mathcal{U}$, the time spent in next state $\mathcal{O}$), are independent. For the UCLA traces, the bottom panel of Table 1 presents the percentage of packet streams for which the test disproves the null hypothesis of independence with a confidence of 95%. From the experimental data displayed in Table 1, we conclude that the macroscopic model passes the statistical tests for an overwhelming majority of the investigated streams. Thus, at least as a first-order approximation, the macroscopic model captures TCP dynamics. Of course, this is not really surprising: the dynamics of the TCP congestion window is nonlinear, while our tools (autocorrelation, etc.) are linear. Our result only show that existing correlations (if any) are subtle enough not to be visible using linear statistics.

3 The Microscopic Model

We will now describe the details of the microscopic model. Let us begin our discussion with the packet IDs. Recall that there are two microscopic schemes; one for each state of the macroscopic model. In State $\mathcal{O}$, the ID sequencing is trivial, and apart from knowledge that we are in State $\mathcal{O}$, no additional information is required. In State $\mathcal{U}$, on the other hand, by definition, packets arrive out of order, and structural properties of the reordering events will guide the modeling. More precisely, the basis of our scheme consists of building a dictionary of frequent, well-structured reordering events arising in the observed packet sequences. Consider the example sequence in (1) and its two unordered phases, 4 5 6 3 and 8 9 10 7. These two events are not identical with respect to the number of inversions: three ordered packets precede a fourth packet with lower ID

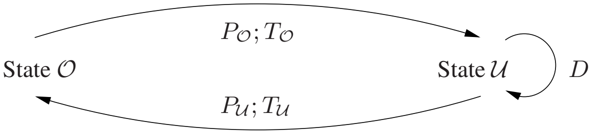

$P_{\mathcal{O}}; T_{\mathcal{O}}$
State $\mathcal{O}$
State $\mathcal{U}$
D
$P_{\mathcal{U}}; T_{\mathcal{U}}$

Fig. 1. Macroscopic model of packet dynamics

THE MACROSCOPIC MARKOV MODEL

Claim I: The sequence of states, $s_1, s_2, \ldots$, is generated according to the following process: Each occurrence of State $\mathcal{O}$ is followed by a number of consecutive occurrences of State $\mathcal{U}$, independently sampled from an underlying distribution D on $\mathbb{N}$.

Claim II: There exist distributions $P_{\mathcal{O}}$ and $P_{\mathcal{U}}$ on $\mathbb{N}$ associated with State $\mathcal{O}$ and State $\mathcal{U}$, respectively, such that for all $n \geq 1$, the number of packets p_n in state s_n is obtained by sampling from the distribution in the set $\{P_{\mathcal{O}}, P_{\mathcal{U}}\}$ that is associated with state s_n.

Claim III: There exist distributions $T_{\mathcal{O}}$ and $T_{\mathcal{U}}$ on $\mathbb{R}_+$ associated with State $\mathcal{O}$ and State $\mathcal{U}$, respectively, such that for all $n \geq 1$, the time t_n spent in state s_n is obtained by sampling from the distribution in the set $\{T_{\mathcal{O}}, T_{\mathcal{U}}\}$ that is associated with state s_n.

Fig. 2. Specification of the macroscopic model

in the first one, while the pattern is more complicated for the second one. However, there is an important way in which the two sequences are similar: if we assume that every packet was ACKed and, furthermore, we employ simple ACKs (not SACK) then *the sequences of ACKs sent in response to receiving those packets are similar*: 3 3 3 7 for the first sequence, 7 7 7 11 for the second one. They are identical modulo a translation in the packet IDs. This motivates the following important notion:

Definition 1. *Two packet sequences A, B are* behaviorally equivalent *(written $A \equiv_{beh} B$) if the sequences ACKs sent in response to receiving the two sequences are identical.*

Behavioral equivalence is a desirable property from the standpoint of TCP modeling: indeed, TCP is a receiver-driven protocol. For behaviorally equivalent traces A and B *the receiver will act identically on A and B.* Assuming similar network conditions, this should make the senders behave in a similar way. So, by regenerating a trace that is behaviorally equivalent to the original trace we are able, indeed, to capture an important part of TCP dynamics.

We will achieve further compression in the microscopic stage by defining a many-to-one mapping from packet sequences in the unordered states to integer "sketches". By the previous discussion, a desired property of this mapping is that behaviorally equivalent sequences are mapped into the same "sketch." Defining a map with this property can be done in several ways, and (as we plan to discuss in a subsequent paper) the right map to use depends on the particular measure of reordering we want to preserve. In this paper we offer a simple solution, achieved by computing the minimum *buffer size*,

Table 1. Percentage of Streams Failing the Statistical Independence Tests

	$N_{\mathcal{U}}$	$P_{\mathcal{O}}$	$P_{\mathcal{U}}$	$T_{\mathcal{O}}$	$T_{\mathcal{U}}$	$P_{\mathcal{O}\to\mathcal{U}}$	$P_{\mathcal{U}\to\mathcal{O}}$	$T_{\mathcal{O}\to\mathcal{U}}$	$T_{\mathcal{U}\to\mathcal{O}}$
TRACE5	0.51	2.71	0.08	2.26	2.00	1.74	2.00	1.72	2.00
TRACE7	1.34	1.80	1.30	4.60	2.30	2.79	3.21	3.00	3.54
TRACE8	1.42	0.30	0.20	0.78	1.10	7.90	8.20	3.15	3.60
TRACE9	2.78	0.90	1.50	1.30	3.40	7.30	7.60	2.78	3.33
TRACE10	5.35	1.70	0.00	1.70	1.50	1.50	1.50	1.00	1.38

denoted MBS, which is the size of the smallest buffer large enough to store all packets that arrive out of order, if we reserve space for not yet received packets. Let us clarify this with a formal definition:

Definition 2. *Let* LOP *be the largest ordered packet ID that has been received (at any given time), i.e., the largest packet ID such that packets in the range* 1 *to* LOP *have all been received (*0 *if packet* 1 *has not yet been received), and let* LRP *be the largest received packet ID (at any given time) (*0 *if no packets have been received yet). We then define the* minimum buffer size (MBS) *as* MBS = LRP − LOP. *Also, the* MBS pattern *associated with a sequence* A *of packet IDs is defined as a time-series of* MBS *values computed* after *each packet in* A *is received. In other words, the* MBS pattern *represents the time evolution of the* MBS.

Returning to our example, we arrive at an identical buffer pattern, $4\ 5\ 6\ 3 \rightarrow 2\ 3\ 4\ 0$, $8\ 9\ 10\ 7 \rightarrow 2\ 3\ 4\ 0$. Since the sequence of packet IDs in State $\mathcal{U}$ always ends with a pivot packet, the last entry in any pattern is 0, and could be omitted in the encoding scheme.

The buffer size in Definition 2 has a natural interpretation in terms of TCP semantics: When all packets have the same payload p, MBS is linearly related to the size of the advertised window, $\mathrm{AdvertisedWindow} = \mathrm{MaxRcvBuffer} - p \cdot$ MBS, where $\mathrm{AdvertisedWindow}$ and $\mathrm{MaxRcvBuffer}$ are defined in [19]. Having introduced a simple encoding scheme for the reordering events, a decoding map can be computed just as easily, because of the following result

Proposition 1. *Consider an* MBS *pattern* B *that consists of* N *positive integers,* $B = (B_1, B_2, \ldots, B_N)$*, and denote by* $B_{\max}$ *the largest integer in* B*. There exists an integer* C *and an algorithm of complexity polynomial in* $N + B_{\max} + C$ *that takes* B *as input and (i) decides whether there exists a corresponding reordering pattern,* $A = (A_1, A_2, \ldots, A_N)$*, and (ii) computes such a pattern if one exists. The smallest integer in* A *is* $C + 1$.

Because of space constraints we omit the mathematical proof of Proposition 1 (see companion paper [20] for details). Consider now the following notion of equivalence between TCP traces: Two sequences of packets A and B are MBS *equivalent* (written $A \equiv_{buf} B$) if the sequences of values of buffer sizes MBS corresponding to the two sequences are identical. We can now restate Proposition 1 as follows: the polynomial time algorithm from Proposition 1 produces ID sequences that are MBS equivalent to the original one. Proposition 1 also guarantees the semantic similarity of the regenerated ID sequences to the original ones. The reason is the following result:

Proposition 2. *[20] Suppose that the receiver uses simple ACKs and acknowledge every packet. Then any sequences* A *and* B *that are* MBS *equivalent are also behaviorally equivalent.*

4 Regeneration of Synthetic TCP Connections

The algorithm in the previous section allows us to regenerate a sequence of packet IDs that provides a significant compression with respect to the original sequence, while being also reasonably plausible as a sequence of received IDs. Indeed, in the ordered state the sequence of packet IDs is increasing, mirroring the increase of the congestion window, while in the unordered state the sequence corresponds to a reordering pattern that occurred in the real trace. The drawback of the model so far is that it does not include regeneration of the *arrival times*. In this section we show how to transform the model into one that jointly generates packet IDs and arrival times. The higher level of the regeneration algorithm will run the Markov chain whose parameters were inferred in the learning phase. This allows regeneration of both the sequence of packet IDs (as detailed in the previous section) and of the time spent by the sequence in each state. On the other hand, we now have to "fill in the details," by assigning arrival times to the regenerated packets.

PACKET ID ALGORITHM

Learning Phase

1. In State $\mathcal{O}$, learn the distribution $P_{\mathcal{O}}$.

2. In State $\mathcal{U}$, instead of learning the distribution $P_{\mathcal{U}}$, learn the *distribution of reordering patterns*, P_{MBS}.

Regeneration Phase

1. In State $\mathcal{O}$, sample from $P_{\mathcal{O}}$ and generate a corresponding number of ordered packet IDs.

2. In State $\mathcal{U}$, sample from P_{MBS} and use the inverse mapping from Proposition 1 to generate a corresponding sequence of IDs.

PACKET ARRIVAL TIME ALGORITHM

Learning Phase

1. Learn the distributions D_{inter} and D_{intra} of inter- and intra-cluster times.

2. Learn the rate r of the best-fit exponential distribution for cluster length in State $\mathcal{U}$.

Regeneration Phase

1. Use D_{inter} and D_{intra} to generate inter-cluster and intra-cluster times, respectively.

2. Simulate the finite state machine from [21] to keep track of the value of the congestion window CWND.

3. In State $\mathcal{O}$: group packets into clusters according to the additive-increase mechanism of TCP and the values of p and CWND.

4. In State $\mathcal{U}$: group packets into clusters assuming an exponential distribution of clusters with rate r.

Fig. 3. Algorithms for learning and regeneration of packet IDs and arrival times

It is important to realize that *by including time spent in various states in the details of the macroscopic model we are already able to capture some temporal characteristics of TCP traffic*. To substantiate this statement we first discuss a really naive approach for arrival time reconstruction. As we show in Section 4.1, even this approach is, however, good enough to recover some basic measures of QoS, such as connection *throughput*. We will next refine the approach, using the results of [21], in order to further incorporate some of the semantical aspects of TCP dynamics. It is fairly clear that even on the receiver side inter-packet times are not random, but display significant correlations. Indeed, an inspection of the data reveals the fact that packets arrive in *clusters*, that are correlated with the dynamics of the congestion window. Being able to group packets in clusters allows us to divide inter-packet arrival times into *intra-* and *inter-cluster* times. We will further make the simplifying assumption that *inter- and intra-cluster times are independent samples from a given distribution, one for each of the two types of times*. Thus, in the learning phase we infer distributions D_{inter} and D_{intra} of inter- and intra-cluster times, respectively. The naive approach we implemented is to *assume that cluster sizes are exponentially distributed*. In the learning phase we infer the parameter of the exponential distribution, used in the regeneration phase to decide whether the next packet is from the same or from a different cluster. This approach does not capture the dynamics of the congestion window, but the requirement that the learning/regeneration algorithms be executable on-the-fly severly limits the range of approaches we can take. Also, as demonstrated by Section 4.2, at least some QoS measures are well captured by this approach. This naive approach is conceptually simple, and easy to implement. It is likely to not be adequate for "microscopic" measures. In particular one should not expect to capture packet clustering, and measures of QoS (e.g. *jitter*) that depend on it.

One can refine the model to a certain extent to further capture aspects of TCP semantics without making very specific assumptions about network influence on TCP dynamics. The refined model will share the same macroscopic features with its simpler version (in particular it will recover throughput just as well as the simple one). In order to regenerate meaningful traces the algorithm will track the value of the congestion window of the trace regenerated so far. This has been accomplished by a heuristic from [21] that we incorporate in our approach. Applying this algorithm provides a good solution to the clustering problem in the ordered phase of RESTORED. Indeed, *in this phase it is reasonable to assume that packets sent in a cluster arrive together as a cluster and in the same order*. Given the additive increase congestion-control mechanism of TCP, this assumption is enough to group packets into clusters that correspond to receiving a whole congestion window. The connection between clustering and the dynamics of the congestion window is only valid in the absence of congestion (in the ordered phase). In the presence of reordering, packet drops and repeats, and faced with potentially different ACK mechanisms, no principled solution seems entirely natural without further assumptions on reordering. Therefore, for the unordered phase we employ the simpler approach outlined previously. The regeneration of arrival times thus takes the form displayed in the bottom panel of Fig. 3.

4.1 Compression and Regeneration Performance

In this section we first present results concerning the performance of RESTORED in compressing traces, followed by comparisons of original vs. regenerated traces with respect to four measures of QoS: throughput and three reordering metrics. Table 2 presents compression ratios for the five TCP traces recorded at UCLA, as well as for the long TCP trace generated by NS-2. Since we have chosen to store the reordering patterns of the observed packet IDs using semantic, *lossless* compression, it is interesting to first see how well RESTORED is able to compress the ID part (before also considering times, for which we have suggested a more compact description). The third column of Table 2 lists the ratio between the size of the original ID sequences and the size of the dictionary (which stores encoded patterns and their associated frequencies).

It can be seen that even this simplistic framework is able to provide decent compression. For aggregate data the compression ratio should be even better due to the succinct modeling of arrival times. To boost the overall compression we prune the dictionary by simply removing extremely infrequent codewords. This way we arrive at the ratios listed in the fourth column of Table 2. In the general case, pruning of the dictionary can be done online using techniques for dealing with *iceberg queries*, well-documented in the database literature.

4.2 Recovering Throughput

We ran the implementation of our naive algorithm on the five sets of real-life connections. For each connection C in one of the traces we reconstructed one sample connection $R(C)$ and computed the throughput ratio, defined as $Q(C) = Throughput(C)/ Throughput(R(C))$. For perfect throughput recovery the value of $Q(C)$ should be 1. In Table 3 we display the concentration of throughput ratios $Q(C)$ around the ideal value 1.

Despite using a method that discards a lot of information (achieving compression rates of the order of 10,000), in most cases our throughput estimates differ by no more than 10% from the true values of the throughput. We feel this is acceptable, given the stringent compression requirements of our method, and has the potential of further being improved if we incorporate some of the *nonlinear* correlations present in TCP.

4.3 Recovering Reordering Metrics

The *inversion spectrum* of a trace is simply the distribution of inversions in the observed reordering patterns. For two packet IDs p_i and p_j with associated indices i and j, we define an inversion as $p_i - p_j$ if $p_i > p_j$ and $i < j$. For example, the sequwnce of packet IDs 2 3 3 1 has one inversion of size 1 $(= 2-1)$ and two inversions of size 2 $(= 3-1)$. Of course, this is not the only way to characterize inversions; an important alternative (see e.g. [10]) is to consider the probability that two packets sent at a time difference of Δt will be received out of order. However, this definition of inversions does not take into account the fact that the sender rate varies. Our definition is also well-suited for a *receiver-oriented* view of network traffic, since it is the relative order of the received packets (and not their temporal lag) that will determine the information in the next ACK packet. Although pruning the dictionary inevitably makes the compression *lossy*, we regenerate synthetic traffic whose inversion structure shows a very high degree of fidelity. This is clear from Fig. 4, in which we have plotted the inversion spectra for NS-2-LONG and two RESTORED reconstructions.

Table 2. Compression Ratios

	TRACE5	TRACE7	TRACE8	TRACE9	TRACE10	NS-2-LONG
Trace size	253MB	247MB	257MB	261MB	119MB	15.4GB
Pattern compression w/o pruning	528×	254×	813×	318×	1 164×	3 366×
Overall compression w/ pruning	16 443×	15 200×	16 542×	16 279×	11 709×	1 079 862×

Table 3. Original vs. Reconstructed Throughput

Throughput ratio $Q(C)$	TRACE5 %	TRACE7 %	TRACE8 %	TRACE9 %	TRACE10 %
0.95–1.05	65.8	68.1	64.2	64.0	63.7
0.90–1.10	85.8	85.5	81.6	83.4	80.3
0.85–1.15	94.4	93.1	91.7	92.3	91.9
0.80–1.20	97.9	97.1	96.2	96.5	96.5
0.75–1.25	98.8	98.1	97.9	97.8	97.9

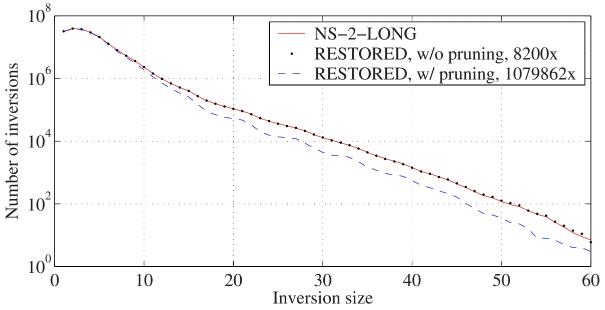


Fig. 4. Inversion spectrum of NS-2-LONG vs. RESTORED

Table 4. Recovering Reordering Metrics

	SUS				RD					RBD				
	2	3	4	5	-2	-1	0	1	2	0	1	2	3	4
TRACE5	95.07	4.54	0.33	0.04	0.20	0.78	98.20	0.23	0.12	98.59	0.46	0.29	0.20	0.13
RESTORED	95.10	4.51	0.34	0.04	0.20	0.78	98.23	0.23	0.10	98.62	0.45	0.27	0.19	0.13

As a second quantitative measure of disorder, we chose to compute shuffled upsequences (SUS) [22]. This measure is defined as *the minimum number of ascending subsequences into which we can partition each listed sequence of packets.* If we compute the SUS metric for *all* sequences of packet IDs corresponding to the unordered state we find that over 95% of them are of type SUS $=$ 2. In fact even packet sequences with a considerable number of inversions are relatively ordered with respect to the SUS measure. This motivates our choice of SUS, as a metric reasonably orthogonal to the distribution of inversions captured by the inversion spectrum. We compare the distribution of SUS values of sequences produced by RESTORED (as a with those of connections in one of the real-life traces. The results are presented in Table 4, and the conclusion is that traces produced by RESTORED are very similar to the original ones (with respect to the SUS measure).

Finally two reordering metrics, introduced by Jayasumana *et al.* are *Reorder Density* and *Reorder Buffer-Occupancy Density (RBD)* [23, 24]. These measures are based on a notion of packet *displacement*. In the interest of space we point the reader to [23, 24] for precise definitions. We have computed the RD metric, aggregated over all connections in one trace, for both one of the original traces, and the corresponding sequences produced by RESTORED. Table 4 presents a comparison between these two distributions, tabulated for the most frequent values of displacement. As we can see the agreement is very good. Similar results hold at the individual connection level, as well as for the other real-life traces. Also, Table 4 presents the distribution of RBD, aggregated over all connections in Trace5, as well as the one from regenerated traces. As we can see, we are able to recover RBD distribution very well.

Acknowledgments. This work has been supported by the U.S. Department of Energy under contract W-705-ENG-36. Special thanks to Hot Rocks Café in Los Alamos.

References

1. Cisco Systems netflow, Available at http://www.cisco.com/warp/public/732/Tech/netflow.
2. W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, Addison-Wesley, 1999.
3. C. Cranor et al. "Gigascope: A stream database for network applications," in *Proc.SIGMOD* 2003, 647–651.
4. W. Leland et al., "On the self-similar nature of Ethernet traffic (extended version)," *ACM/IEEE Transactions on Networking*, vol. 2 (1), pp. 1–15, 1994.
5. R. H. Riedi et al., "A multifractal wavelet model with application to network traffic," *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 992–1018, April 1999.
6. D. Figueiredo et al. "On TCP and self-similar traffic," *J. Perf. Evaluation (to appear)*, 2005.
7. J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "Internet traffic tends to Poisson and independent as the load increases," Bell Labs, Murray Hill, NJ, Tech. Rep., 2001.
8. D. Figueiredo et al. "On the autocorrelation structure of TCP traffic," *Computer Networks*, vol. 40, no. 3, pp. 339–361, October 2002.
9. J. Bennett et al., "Packet reordering is not pathological network behavior," *IEEE/ACM Transactions on Networking*, vol. 7 (6), pp. 789–798, 1999.
10. J. Bellardo and S. Savage, "Measuring packet reordering," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, Nov. 2002, pp. 97–105.
11. W. Fang and L. Peterson, "Internet-AS traffic patterns and their implications," in *Proc. IEEE GLOBECOM Conf.*, 1999, pp. 1859–1868.
12. A. Feldmann et al. "Deriving traffic demands for operational IP networks: Methodology and experience," in *Proc. ACM SIGCOMM* , 2000, pp. 257–270.
13. C. Estan and G. Varghese, "New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice," *ACM TOCS*, vol. 21 (3), pp. 270–313, 2003.
14. K. Park and W. Willinger, Eds., *Self-similar network traffic and performance evaluation*. Wiley, 2000.
15. J. Cao et al. "On the nonstationarity of Internet traffic," in *Proc. ACM SIGMETRICS*, 2001.
16. P. Brockwell and R. Davis, *Introduction to Time Series and Forecasting*, Springer, 2002.
17. G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976, p. 33.
18. R. Hogg and A. Craig, *Introduction to Mathematical Statistics*, Macmillan, 1995.
19. L. Peterson and B. Davie, *Computer Networks. A Systems Approach*, M. Kauffman, 2000.
20. A. Hansson, G. Istrate, and S. Kasiviswanathan, "Combinatorics of TCP reordering," submitted to a special issue of *Journal of Combinatorial Optimization*, July 2005.
21. S. Jaiswal et al. "Inferring TCP connection characteristics through passive measurements," in *Proc. IEEE INFOCOM.*, 2004.

22. V. Estivill-Castro and D. Wood, “A survey of adaptive sorting algorithms,” *ACM Computing Surveys*, vol. 24, no. 4, pp. 441–476, December 1992.
23. A. Jayasumana et al. “Reorder density and reorder buffer-occupancy density—metrics for packet reordering measurements”, IETF IP Performance Metrics WG, Available at http://www.ietf.org/internet-drafts/draft-jayasumana-reorder-density-04.txt.
24. A. Jayasumana et al., “RD: A formal, comprehensive metric for packet reordering,” in *Proc. IFIP Networking*, 2005.
25. A. Veres and M. Boda, “The chaotic nature of TCP congestion control,” in *Proc. IEEE INFOCOM* 2000, pp 1715–1723.

Traffic Anomaly Detection and Characterization in the Tunisian National University Network

Khadija Houerbi Ramah[1], Hichem Ayari[2], and Farouk Kamoun[2]

[1] Ecole d'Aviation Borj El Amri
khadija.houerbi@cristal.rnu.tn
[2] CRISTAL laboratory,
École Nationale des Sciences de l'Informatique,
University of Manouba,
2010 Manouba, Tunisia
H.ayari@ensi.rnu.tn, farouk.kamoun@ensi.rnu.tn

Abstract. Traffic anomalies are characterized by unusual and significant changes in a network traffic behavior. They can be malicious or unintentional. Malicious traffic anomalies can be caused by attacks, abusive network usage and worms or virus propagations. However unintentional ones can be caused by failures, flash crowds or router misconfigurations. In this paper, we present an anomaly detection system derived from the anomaly detection schema presented by Mei-Ling Shyu in [12] and based on periodic SNMP data collection. We have evaluated this system against some common attacks and found that some (Smurf, Sync flood) are better detected than others (Scan). Then we have made use of this system in order to detect traffic anomalies in the Tunisian National University Network (TNUN). For this, we have collected network traffic traces from the Management Information Base MIB of the central firewall of the TNUN network. After that, we calculated the inter-anomaly times distribution and the anomaly durations distribution. We showed that anomalies were prevalent in the TNUN network and that most anomalies lasted less than five minutes.

Keywords: Anomaly Detection, Principal Component Analysis, Temporal Characteristics.

1 Introduction

For the last few years, we have observed a continuous increase of malicious traffic in the Internet in form of distributed denial of service attacks, virus and worms propagation, intrusions, etc. In fact recent studies ([4], [8], [14]) have revealed the important rise in malicious traffic volume in the entire Internet. This rise is in a huge proportion caused by the propagation in the Internet of worms such as CodeRed [5] [13], Nimda [13], the Slammer worm [6], Msblaster and Funlove. Consequently, defending networks against such malicious traffic is a day by day incessant activity for network operators.

A lot of techniques have been developed in order to detect, identify and prevent propagation of malicious traffic over networks. We differentiate between two classes

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 136 – 147, 2006.

of intrusion detection techniques: Misuse Detection and Anomaly Detection. The Misuse Detection Systems try to detect intrusions by comparing the current activity of the audited resource to a database of known attack scenarios. Those techniques can not detect unknown attacks. However, the Anomaly Detection Systems (ADS) try to detect intrusions by comparing the current activity of the audited resource to an established "normal activity" represented in form of a profile.

The majority of these techniques need to keep per-connection or per-flow state over a single link or node. Thus, they must be widely deployed in all nodes in order to be effective. Moreover they require a lot of computing resources making their cost unaffordable for many ISPs. In this work, we tried to develop an anomaly detection tool able to detect attacks without keeping a per flow state. This tool doesn't attempt to identify the different types of attacks or their origins. So it can be useful as a first-line anomaly detection tool. In fact, this tool can be used to indicate when a more sophisticated intrusion detection system, based on per-flow data collection, must be started.

In fact, we developed an Anomaly Detection System (ADS) derived from the anomaly detection schema presented by Mei-Ling Shyu in [12] and based on SNMP data. After evaluating this system against some common attacks, we exploit it for the detection of traffic anomalies in the TNUN network. Finally we studied some temporal patterns of network traffic anomalies.

This paper is organized as follow. First, in the second section, we discuss previous related work. In the third section, we describe the anomaly detection technique used by our ADS system. Then we present the evaluation method and discuss evaluation results. In section four we describe the TNUN network. After that, we discuss some temporal characteristics of traffic anomalies in the TNUN network in the fifth section. Finally we conclude with a summary of the themes developed during our study.

2 Related Work

Anomaly detection techniques always start by the construction of a profile for "normal" network behaviour and then mark deviations from such profile as possible attacks. Many approaches have been proposed since anomaly detection was originally proposed by Denning in [7] and they are mainly statistical ones. Indeed, the definition of a normal profile, in those approaches, relies on the use of known statistical properties of normal traffic or on a training period. Then those approaches employ statistical tests to determine whether the observed traffic deviate significantly from the norm profile. The work of J Brutlag in [2] and the one of R Kompella in [16] are examples of such statistical approaches.

Some other statistical approaches are based on clustering techniques ([3], [11], [12]). For example, in [11], Chhabra presents an algorithm that monitors packets at network components and uses a clustering technique to group active flows into categories based on common values in the fields of the packets. If the total number of packets in a cluster is greater than a specified threshold, then the common fields and the corresponding values for the packets in the cluster form an attack signature.

On the other hand, anomaly characterization is the subject of recent research aiming at understanding anomalies statistical, temporal or spatial behaviour in order to be able to develop better and more powerful ADS in the future. Some anomaly characterization studies were based on identified attack traces ([1], [14]). For example, the study elaborated by Yegneswaran in [14] was based on intrusion logs from firewalls and IDS systems at sites distributed throughout the Internet. However, Pang anomaly characterization in [8] was based on measuring background radiation (traffic sent to unused or unallocated IP addresses). Several other studies were based on anomaly traces generated by previously implemented ADS ([3]). All these studies have showed some interesting characteristics of anomalies.

In fact, in [1] Barford used SNMP data, IP flow data and a journal of known anomalies and network events in order to achieve wavelet analysis of network traffic anomalies. He classified anomalies into three groups: network operation anomalies, flash crowd anomalies and network attack anomalies. He found that flash crowd events were the only long lived anomaly events. He also showed that coarse-grained SNMP data can be used to expose anomalies effectively.

By analysing a set of firewall logs, the authors in [14] found that the Internet suffers from a large quantity and wide variety of intrusion attempts on a daily basis. They also found that the sources of intrusions are uniformly spread across the Autonomous System space. The authors affirmed also that a very small collection of sources are responsible for a significant fraction of intrusion attempts in any given month and their on/off patterns exhibit cycles of correlated behaviour. They also found that worms like codeRed or Nimda persist long time after their original release. Finally, they established that the distribution of source IP addresses of the non-worm intrusions as a function of the number of attempts follows Zipf's law.

In [8], the authors used traffic filtering and honeypots techniques in order to study the characteristics of "background radiation" (traffic sent to unused addresses). They broke down the components of this non-productive traffic by protocol, application and often specific exploits, they analysed temporal patterns and assessed variations across different networks and over time. They found that worms probes and "autorooter" scans (similar to worms, but without self propagation) heavily dominate background radiation.

In [3], the authors found that the anomalies are highly diversified including denial of service attacks, flash crowds, port scanning, downstream traffic engineering, high-rate flows, worm propagation and network outages. They also found that most anomalies are small in time (duration) and space (Number of Origin-Destination flows implicated in each anomaly).

3 The Anomaly Detection System

In order to detect anomalies we developed an ADS tool based on the work of Shyu in [12]. In fact, in [12], Shyu proposed an unsupervised anomaly detection schema based on Principle Component Analysis (PCA) and assuming that anomalies can be detected as outliers.

PCA is a multivariate method, concerned with explaining the variance-covariance structure of a set of variables through a few new variables which are linear combinations of the original ones. On the other hand, outliers are defined as observations that are different from the majority of the data or are sufficiently unlikely under the assumed probability model of data [12].

Shyu has evaluated her method over the KDD CUP99 data and she has demonstrated that it exhibits better detection rate than other well known outlier based anomaly detection algorithms such as the Local Outlier Factor "LOF" approach, the distance of Canberra based approach, the Nearest Neighbour approach and the Kth Nearest Neighbour approach.

KDD CUP99 data is the data set used for the Third International Knowledge Discovery and Data Mining Tools Competition. It is composed of TCP connection records labelled as either normal or as an attack with one attack type.

In our ADS tool we propose to use SNMP data. Although this information gives us an aggregated view of the state of the network traffic, it has the advantage to be simple, consume acceptable amount of resources and so it can be used for real time anomaly detection. So we choose to collect the following "MIB" counters for any given monitored equipment: ifInUcastPkts (number of received unicast packets by an interface), ifInOctets (number of received octets by an interface), ifOutUcastPkts (number of unicast packets send by an interface) and IfOutOctets (number of octets transmitted by an interface).

We have implemented this ADS tool using MATLAB environment. For the collection of SNMP data, we used a commercial network management system Whats UP [15].

In the Next section we present the Shyu's anomaly detection schema used by our ADS tool.

3.1 Shyu's Anomaly Detection Schema

Shyu's method needs, to perform PCA, a robust estimation of the correlation matrix and the mean of the normal observations. In order to obtain such estimators, from a data set of unsupervised data, Shyu proposes the use of the multivariate trimming technique based on the Mahalanobis distance in order to identify the β% (β is given) extreme observations that are to be trimmed. The Mahalanobis distance is calculated as in Eq. 1 for each observation x_i.

$$d_i^2 = \left(x_i - \bar{x}\right)' S^{-1}\left(x_i - \bar{x}\right) \tag{1}$$

Where $\bar{x}$ is the arithmetic mean estimator and **S** is the correlation matrix estimator.

Subsequently, the robust estimators of arithmetic mean and the correlation matrix are calculated from the remaining observations.

In Shyu's method, PCA analysis is based on the use of both major principle components and minor ones, in order to detect both outliers with respect to one variable and multivariate outliers. For the selection of these principle components, Shyu proposes to select the q major principal components that account for a given amount of energy (for example: 50 % of total data set energy). For the minor ones, she proposes to choose them from principal components which eigenvalues are less than to 0.20.

Given the q major and r minor components selected from p principal components, an observation x is classified as an attack if it satisfies Eq. 2, otherwise it is classified as normal.

$$\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} > c_1 \quad \text{or} \quad \sum_{i=p-r+1}^{p} \frac{y_i^2}{\lambda_i} > c_2 \tag{2}$$

Where y_i is the i^th^ principal component and λ_i is the corresponding eigenvalue. c_1 and c_2 are outlier thresholds determined according to the classifier specified false alarm rate.

3.2 Evaluation Method

In order to evaluate our ADS, we need a trace where traffic anomalies are well identified. So we have deployed an experimental network (Figure 1) which consists of two local networks connected by a router. In order to simulate normal traffic, we used a network traffic generator LANTRAFFIC [9] which maintains sixteen TCP and UDP bidirectional connections between a victim and the traffic generator machine. Those connections are completely customizable (data length, time between packets, connection generation distribution, packets length…). We also deployed two machines in order to launch attacks over this experimental network. Finally, we

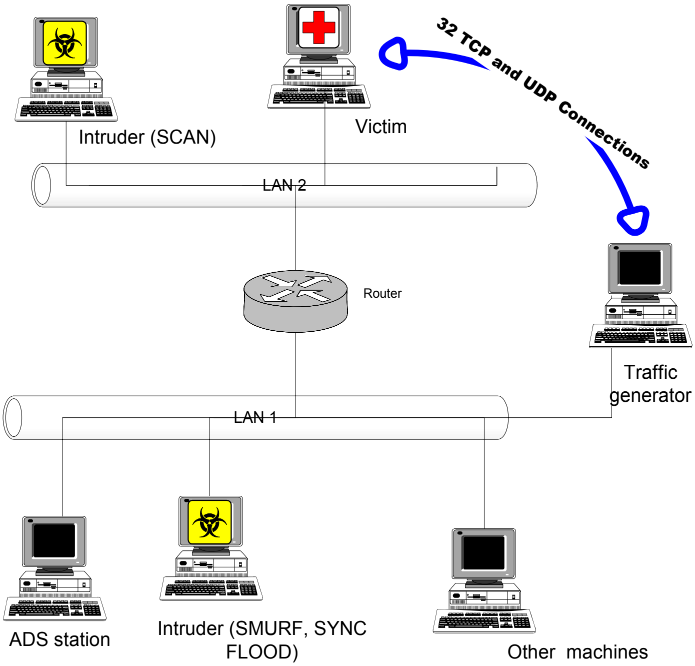


Fig. 1. The experimental network for the ADS evaluation

deployed our ADS station which processed the collected SNMP data from the central router. This data was collected by a WhatsUP management system every 20 seconds.

For the ADS system we fixed the amount of energy explained by the chosen major principal components to be at least equal to 50% of total data set energy and the trimming to be 0.5% of all observations in the data set.

We evaluate our ADS tool according to the following general and per-attack metrics presented by Lazarevic in [10] (tables 1 and 2).

Table 1. General metrics definition

Real false alarm rate	Number of false alarms divided by the total number of observations
Detection rate	Number of truthful alarms divided by the total number of real anomalous observations
Precision	Number of truthful alarms divided by total number of alarms

Table 2. Per-attack metrics definition

Burst Detection Rate (bdr)	Ratio between total number of intrusive observations that have score value higher than threshold and the total number of real intrusive observations
Response Time (Trep)	Time elapsed from the beginning of the attack until the moment the score value reaches the threshold

3.3 Evaluation Results

Three different types of attacks were launched from the two intruder machines at fixed moments illustrated in figure 2. The chosen attacks are SMURF, SYN-Flood and a network scan attack performed by the NMAP tool. The first two attacks are Deny Of Service (DOS) attacks using flooding techniques. In figure 3, we show the repartition over time of anomalies detected by our ADS.

When we increase the fixed false alarm rate, the precision decrease rapidly but the detection rate didn't greatly improve (Table 3). In fact, a fixed false alarm rate of 2% offers acceptable performance.

We remark also, that some attacks are better detected by our system than others (Table 4). In fact, Smurf and SYN-flood attacks are precisely detected (burst detection rate near 100%) and rapidly (response time near 0). Furthermore, we remark that network scan is difficult to detect (burst detection rate very low) and need more time for detection. We think that this low detection rate of network scan anomalies is due to the fact that we have used only one scan process in our experimentation. However in real networks, we assist nowadays to a continuous apparition of new worms and virus that start multiple network scanning threads in each infected machine in order to find backdoors and security holes in other computers. If one vulnerable computer is detected, those worms copy themselves in the victim system which also starts scans in order to attack other computers. So, we think that the impact of those scanning activities will be more apparent in the case of real worm infection than it was in our experiment and we expect to have a better detection rate of our algorithm.

The performances of our ADS tool are not as good as those obtained by Shyu in [12]. In fact, she obtained 98.94% for detection rate and 97.89% for precision with a

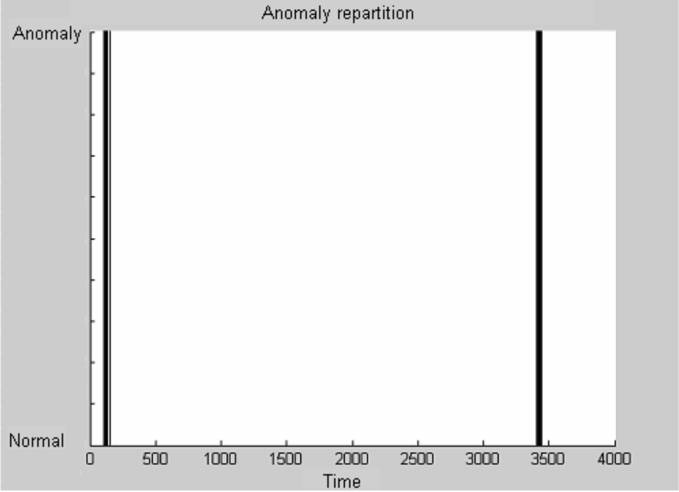


Fig. 2. Real anomaly repartition over time

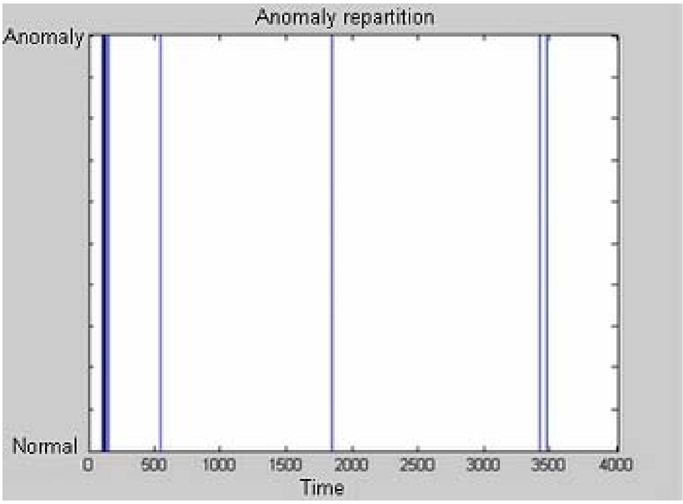


Fig. 3. detection results for a 2% fixed false alarm rate

false alarm rate of 0.92%. But, we must notice that Shyu used her method in a supervised manner. In fact, all outlier thresholds were determined from a training data composed of 5000 normal connections.

In our case, we used our ADS tool with no training period, because in real networks it's very difficult to have a training period composed of only normal traffic. Moreover, our ADS tool is simpler than Shyu's method because it is based only on SNMP data (8 variables in this evaluation test).

Table 3. Variation of the general performances according to the fixed false alarm rate

Fixed false alarm rate	2%	4%	6%
Observed false alarm rate	1,14%	1,71%	2,54%
Detection rate	47,14%	62,86%	68,57%
Precision	91,67%	56,41%	41,74%

Table 4. Variation of performances by attack type according to the fixed false alarm rate

	Fixed false alarm rate					
	2%		**4%**		**6%**	
	bdr	**Trep**	**bdr**	**Trep**	**bdr**	**Trep**
Smurf	0,93	1	0.97	0	0,97	0
SYN flood	1	0	1	0	1	0
SCAN	0.03	19	0.32	19	0.44	3

Whereas Shyu's method is based on per-flow data (TCP connections composed of 41 variables). In addition, the size of SNMP data used by our ADS tool depends only on the period of collection; whereas the size of the data used by Shyu's method and other ADS systems based on per-flow data depends on traffic volume which makes these systems difficult to adapt for real time anomaly detection in high speed networks.

4 TNUN Network

After evaluation of the ADS tool, we used it in order to detect anomalies in the Tunisian National University Network (TNUN).

The TNUN network is connecting all Tunisian universities to each others and to the Internet. It is composed by a unique central node located at the region of Tunis / El Manar and more than one hundred dispersed universities. In fact, all universities institutions are connected to this central node by mean of direct leased lines or indirect ones (throw the Tunisian national backbone). This central node treats all the network traffic between universities and the Internet and is designed around a central firewall (Fig 4). Thus, the central firewall represents the ideal point of data collection.

So In order to detect anomalies in the TNUN network, we collected periodically, every minute, “MIB” information counters from this central firewall.

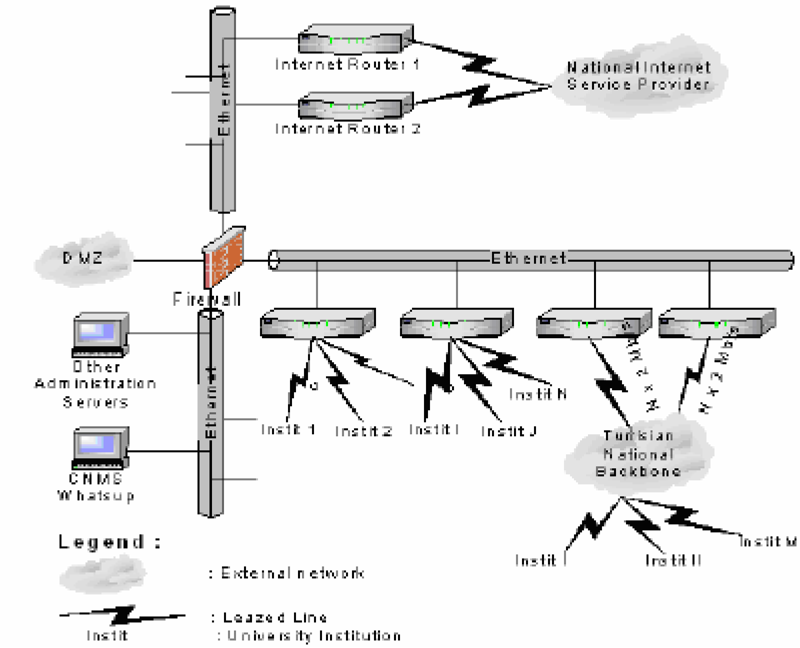


Fig. 4. The TNUN Network: CCK/ EL Manar Central Node

5 Characterization of Anomalies in TNUN

In order to study network anomaly characteristics, we define two temporal metrics. The anomaly duration is the lapse of time during which all samples are labeled as anomalous by the ADS system. The inter-anomaly time is the time between the end of an anomaly and the beginning of the next one.

We used the ADS to detect anomalies in TNUN network, for a 45 days period (between 03/04/2004 and 18/05/2004).

We found that anomalies are frequent in the TNUN network. In fact, figure 5 shows that more than 50% of anomalies are separated by less than 60 minutes. We also found that most anomalies are short lived. In fact, figure 6 shows that 90% of anomalies last less than 5 minutes.

These results are consistent with previous studies witch have established that attacks are very frequent in the Internet. For example in [8], Pang affirmed that in the Lawrence Berkeley National Laboratory (LBL), in one arbitrarily-chosen day, about 8 millions connection attempts are scans. This number account for more than double the site’s entire quantity of successfully established incoming connections. In [3] Lakhina affirmed that anomalies can last anywhere from milliseconds to hours and that the most prevalent anomalies in his datasets are those that last less than 10 minutes.

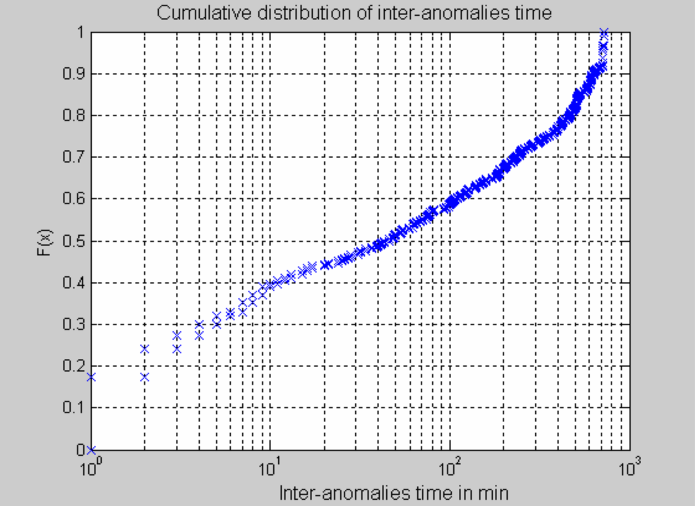


Fig. 5. Cumulative Distribution of inter-anomalies time

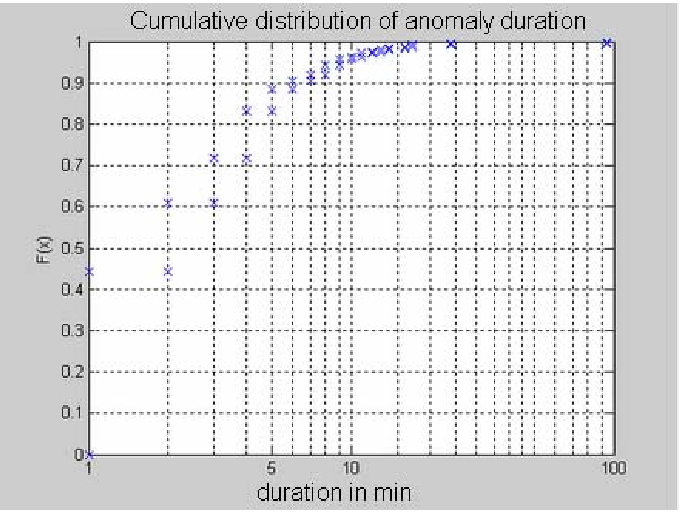


Fig. 6. Cumulative Distribution of anomaly duration

6 Conclusion

In this paper, we presented a first level anomaly detection system based on SNMP data. This system can be used for automatic real time detection of traffic anomalies.

We have evaluated this system against some well known attacks and found that it is efficient in detecting flooding attacks that disrupt network traffic. These attacks are very difficult to detect with usual intrusion detection systems and to prevent with firewalls because they make use of normal connection attempts.

Next, we showed that in the TNUN network anomalies are prevalent but most of them are short lived. Similar results were previously found in other studies mainly in [3] and [8]. So, we can say that our study offers another proof of the high prevalence of anomalous traffic in Internet.

Finally, we plan to deploy our system over the entire TNUN in order to help network operators in the Tunisian universities early detect on-going attacks. In future work we plan to add to our system modules for attack identification. So network operators can implement filters to mitigate the effect of anomalous traffic on the "good" traffic.

Acknowledgement

This work couldn't be achieved without the active cooperation of the Khawarizmi Calculus Center (CCK). We would like to thank all the CCK personal and particularly his director Madam Henda BEN GHAZALA.

References

1. P. Barford and D. Plonka, Characteristics of Network Traffic Flow Anomalies, in Proceedings of ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA, November 2001.
2. J.Brutlag, "Aberrant Behaviour Detection in Time Series for Network Monitoring", in Proceeding of the USENIX Fourteenth System Administration Conference LISA XIV, new Orleans, LA, December 2000
3. Anukool Lakhina, Mark Crovella, Christophe Diot, Characterisation of Network-Wide Anomalies in Traffic Flows, IMC'04, Italy, October 2004.
4. D. Moore, G. Voelker, and S. Savage: Inferring Internet Denial of Service activity. In Proceedings of the 2001 USENIX Security Symposium , Washington DC, August 2001.
5. D.Moore, C.Shannon and J.Brown: Code-Red: a Case Study on the Spread and Victims of an Internet Worm. In Internet Measurement Workshop (IMW); 2002.
6. D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver: Inside the Slammer Worm. In Security and Privacy, July/August 2003.
7. D.E. Denning: An Intrusion Detection Model. In IEEE Transaction on Software Engineering, 1987.
8. R. Pang, V. Yegneswaran, P. Barford, V. Paxon, L. Peterson: Characteristics of Internet Background Radiation. In IMC'04, Italy, October 2004.
9. LANTRAFFIC : http://www.zti-telecom.com/
10. A. Lazarevic, L. Eroz, V. Kumar, A. Ozgur and J. Srivastava; A Comparative Study of Anomaly Detection Schemes. In Network Intrusion Detection; Proceeding of Third SIAM International Conference on Data Mining; San Francisco; 2003.
11. P. Chhabra, A. John, and H. Saran. "PISA: Automatic Extraction of Traffic Signatures", In fourth International Conference in Networking, Ontario, Canada, May 2005.

12. Mei-Ling Shyu, Shu-Ching Chen, K. Sarinnapakorn, and L. Chang: A Novel Anomaly Detection Scheme Based on Principal Component Classifier. In Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03), pp. 172-179, Melbourne, Florida, USA, 2003.
13. S. Staniford, V. Paxson and N. Weaver: How to 0wn the Internet in Your Spare Time, In Proc. USENIX Security Symposium 2002.
14. V. Yegneswaran, P. Barford and J. Ullrich; Internet Intrusions: Global Characteristics and Prevalence. In SIGMETRICS'03; USA; June 2003.
15. Ipswitch Whatsup CNMS. www.ipswitch.com
16. R. Kompella, S. Singh, G. Varghese: On Scalable Attack Detection in the Network. Internet Measurement Conference 2004: pp 187-2004.

B-EDCA: A New IEEE 802.11e-Based QoS Protocol for Multimedia Wireless Communications[*]

José Villalón, Pedro Cuenca, and Luis Orozco-Barbosa

Albacete Research Institute of Informatics,
Universidad de Castilla-La Mancha,
02071 Albacete, Spain
{josemvillalon, pcuenca, lorozco}@info-ab.uclm.es

Abtract. The IEEE 802.11e draft standard is a proposal defining the mechanisms for wireless LANs aiming to provide QoS support to time-sensitive applications. However, recent studies have shown that the IEEE 802.11e (EDCA) performs poorly when the medium is highly loaded due to the high collision rate. Even though several proposals have been proposed to address this problem, they require important changes to the current standard specifications making difficult their actual implementation. In this paper, we propose a simple QoS-aware mechanism and fully compatible with the various operation modes of the EDCA standard as well as the legacy IEEE 802.11 (DCF) scheme. Our design has been based on an in-depth analysis of the several operation modes of both standards. This should ensure full compatibility of operation: an important feature since the transition from the IEEE 802.11 to the IEEE 802.11e will take some time making more likely the existence of hybrid scenarios where both standards will have to coexist. Our simulation results show that our new scheme outperforms the EDCA and other QoS-aware schemes recently reported in the literature.

1 Introduction

The IEEE 802.11 WLANs [1] is being deployed widely and rapidly in many different environments including enterprise, home and public access networks. One of the most influential factors to its success is due to the development of high-speed technology enabling the deployment of multimedia applications. However, multimedia applications are not only characterized by their high bandwidth requirements, but also impose severe restrictions on delay, jitter and packet loss rate. In others words, multimedia applications require Quality of Service (QoS) support. Guaranteeing those QoS requirements in IEEE 802.11 is a very challenging task due to the QoS-unaware operation of its MAC layer. This layer uses the wireless media characterized by the difficulties faced by the signal propagation. Thus providing QoS to IEEE 802.11 has been and it is an active research area giving rise to numerous service differentiation schemes.

[*] This work was supported by the Ministry of Science and Technology of Spain under CICYT project TIC2003-08154-C06-02, the Council of Science and Technology of Castilla-La Macha under project PBC-03-001 and FEDER.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 148 – 159, 2006.

Currently, the IEEE 802.11 Working Group is hardly working on the definition of the IEEE 802.11e standard [2]. The IEEE 802.11e draft is a proposal defining the mechanisms for wireless LANs aiming to provide QoS support to time-sensitive applications, such as, voice and video communications. The standardization efforts are at their final stage and it is expected that the standard will soon be publicly available.

It is expected that in the near future IEEE 802.11e-compliant interface cards will take over the WLAN market, replacing the use of legacy IEEE 802.11 interface cards in most WLAN applications. The complete migration towards the IEEE 802.11e standard will take several years given the wide scale use of legacy IEEE 802.11 in the market place today. This creates an important number of networking scenarios where legacy IEEE 802.11 based stations and IEEE 802.11e-based stations will have to interwork.

However, the ratification of the IEEE 802.11e standard is becoming a very challenging task. Many studies have shown that the IEEE 802.11e (EDCA) scheme performs poorly under heavy load conditions. The severe degradation is mainly due to high collision rates. This reason has led many researchers to design new techniques aiming to address the shortcomings of the current draft standard. However, many of the proposed techniques have overlooked two main implementation and operation issues: first, the implementation of the proposed mechanisms implies important and incompatible modifications to the IEEE 802.11e specifications in a moment in which IEEE 802.11e is at its final stage, and second, the main deficiency of these mechanisms comes from its inability to provide the QoS guarantees required by the time-constrained flows when legacy DCF based stations are present in the same scenario.

In this paper, we address the two aforementioned issues by introducing an IEEE 802.11e-compliant mechanism capable of providing QoS support even under scenarios where legacy DCF based stations are present. Our main objective has been to design a scheme able to provide the QoS guarantees required by two of the most representative time-constrained multimedia applications regardless of the channel load and under a systems configuration consisting of IEEE 802.11 and IEEE 802.11e-compliant stations. Simulation results show that our new scheme outperforms the IEEE 802.11e draft standard and some of the most relevant schemes reported in the literature. Throughout an exhaustive campaign of simulations, we have evaluated the performance of the system in terms of four metrics: throughput, access delay, delay distribution and packet loss rate.

This paper is organized as follows. Section 2 provides an overview of the IEEE 802.11 WLAN standard. In Section 3, we also describe the upcoming IEEE 802.11e QoS standard and two relevant proposals recently reported in the literature aiming to improve the performance of the IEEE 802.11e standard. In Section 4, we present our new IEEE 802.11e based QoS mechanism. In Section 5, we carry out a comparative performance evaluation when supporting different services, such as, voice, video, best-effort, background and in the presence of traffic generated by legacy DCF based stations. Finally, Section 6 concludes the paper.

2 Overview of IEEE 802.11 WLAN

The IEEE 802.11 MAC sub-layer [1] defines two medium access coordination functions, the *Distributed Coordination Function* (DCF) and the optional *Point*

Coordination Function (PCF). DCF is the basic access function for IEEE 802.11 and is based in a *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) algorithm together with a contention (*backoff*) algorithm. PCF uses a centralized polling method requiring a node to play the role of *Point Coordinator* (PC). The PC cyclically polls the stations to give them the opportunity to transmit. In the following, we restraint our description to the DCF mechanism whose mode of operation may affect the ability of the upcoming IEEE 802.11e (EDCA) standard to provide QoS guarantees.

A station operating under the DCF scheme should first sense the state of the channel before initiating a transmission. A station may start to transmit after having determined that the channel is idle during an interval of time longer than the *Distributed InterFrame Space* (DIFS). Otherwise, if the channel is sensed busy, once the transmission in course finishes and in order to avoid a potential collision with other active (waiting) stations, the station will wait a random interval of time (the *Backoff_Time*) before starting to transmit. As long as no activity is detected in the channel, a backoff counter, initially set to *Backoff_Time*, is decremented on an *aSlotTime* by *aSlotTime* basis. Whenever activity is detected, the backoff counter is frozen and reactivated once again when the channel has remained idle during an interval of time longer than DIFS. The station will be able to begin transmission as soon as the backoff counter reaches zero. In case of an unsuccessful transmission, the station will have a finite number of attempts, using a longer backoff time after each attempt.

Even though DCF is a simple and effective mechanism, DCF can neither support QoS nor guarantee to meet the multimedia applications requirements. It is for this reason that many researchers have proposed techniques the provisioning of QoS mechanisms into the DCF mode of operation. The description of such mechanisms is out of the scope of this work. An overview of many of the different QoS enhancements mechanisms for the IEEE 802.11 standards can be found in [3]. In that work, the authors have summarized and classified a large number of the proposed techniques. A comparative performance evaluation of some of them can also be found in [4], [5], [6].

3 The IEEE 802.11e Draft Standard

The IEEE 802.11e draft standard [2] aims to specify the mechanisms enabling the provisioning of QoS guarantees in IEEE 802.11 WLANs. In the IEEE 802.11e standard, distinction is made among those stations not requiring QoS support, known as *nQSTA*, and those requiring it, *QSTA*. In order to support both Intserv and DiffServ QoS approaches in an IEEE 802.11 WLAN, a third coordination function is being added: the *Hybrid Coordination Function* (HCF). The use of this new coordination function is mandatory for the QSTAs. HCF incorporates two new access mechanisms: the contention-based *Enhanced Distributed Channel Access* (EDCA), known in the previous drafts as the *Enhanced DCF* (EDCF) and the *HCF Controlled Channel Access* (HCCA). In the HCCA mechanism a central node is used for coordinating the access to the channel: the *Hybrid Coordinator* (HC). When the HC takes control over the channel during the *Contention Period* (CP), it is said that a

Controlled Access Phase (CAP) has been generated. It is worth noting that the HC should at all times hold the highest priority allowing it to initiate the CAP.

One main feature of HCF is the definition of four *Access Categories* (AC) queues and eight *Traffic Stream* (TS) queues at MAC layer. When a frame arrives at the MAC layer, it is tagged with a *Traffic Priority Identifier* (TID) according to its QoS requirements, which can take values from 0 to 15. The frames with TID values from 0 to 7 are mapped into four AC queues using the EDCA access rules. The frames with TID values from 8 to 15 are mapped into the eight TS queues using the HCF controlled channel access rules. The TS queues provide a strict parameterized QoS control while the AC queues enable the provisioning of multiple priorities. Another main feature of the HCF is the concept of *Transmission Opportunity* (TXOP), which defines the transmission holding time for each station.

EDCA has been designed to be used with the contention-based prioritized QoS support mechanisms. In EDCA, two main methods are introduced to support service differentiation. The first one is to use different IFS values for different ACs. The second method consists in allocating different CW sizes to the different ACs. Each AC forms an EDCA independent entity with its own queue and its own access mechanism based on an DCF-like mechanism with its own *Arbitration Inter-Frame Space* defined by $AIFS[AC]=SIFS+AIFS[AC]\times SlotTime$ and its own CW[AC] ($CWmin[AC] \leq CW[AC] \leq CWmax[AC]$), where $AIFSN[AC]$ is the *Arbitration Inter Frame Space Number*. If an internal collision arises among the queues within the same QSTA, the one having higher priority obtains the right to transmit. It is said that the queue getting the right to access to the channel obtains a transmission opportunity (TXOP). The winning queue can then transmit during a time interval whose length is given by *TXOPLimit.*

3.1 QoS Enhancements to the IEEE 802.11e

Many on-going research efforts are focusing on the evaluation of the IEEE 802.11e draft standard. Many studies have revealed that the poor performance exhibited by the draft standard is mainly due to the high collision rates encountered when a large number of stations attempt to access the channel. Numerous proposals have been reported in the literature aiming to overcome this main drawback. In the following, we undertake the analysis of two of the most prominent ones.

The *Fast Collision Resolution Mechanism* **FCR** [7] aims to shorten the backoff period by increasing the contention window sizes of all active stations during the contention resolution period. To reduce the number of wasted (idle) slots, the FCR algorithm assigns the shortest window size and idle backoff timer to the station having successfully transmitted a packet. Moreover, when a station detects a number of idle slots (static backoff threshold), it starts reducing the backoff timer exponentially, instead of linearly as specified by the EDCA draft standard. To address the provisioning of QoS mechanisms, the authors further introduce an enhanced version of the FCR algorithm, namely, the *Real Time Fast Collision Resolution* (**RT-FCR**) [7] algorithm. In this algorithm, the priorities are implemented by assigning different backoff ranges based on the type of traffic. In their study, the authors have considered three main traffic types: voice, video, and best-effort (data) traffic.

Under this scheme, voice packets hold the highest priority to access the channel by setting CW = CW_{min}. All the other flows have to wait, at least, eight backoff slots before being allowed to gain access to the channel. The video traffic is assigned the second highest priority by using a smaller maximum contention window size than the one assigned to the best-effort data traffic.

The *Adaptive EDCF Mechanism* (**AEDCF**) [8] is another relevant mechanism recently reported in the literature. In [8], the authors state that the probability of collision increases is due to the re-setting of *CW[AC]* to *CWmin[AC]* after a successful transmission in the presence of multiple stations contending for the channel. Taking this fact into account, they have proposed decreasing the *CW[AC]* by multiplying by a factor lower than 0.8 after a successful transmission; the actual value of the factor will depend on the collision rate suffered by the AC. In [9], the same authors go a step further by introducing a new scheme called *Adaptive Fair EDCF* (**AFEDCF**) that improves AEDCF and FCR mechanisms. This mechanism uses an adaptive fast collision resolution mechanism (similar to the FCR mechanism) when the channel is sensed idle. In contrast with the FCR mechanism, AFEDCF computes an adaptive backoff threshold for each priority level by taking into account the channel load.

However, the main deficiency of these mechanisms comes from its inability to provide the proper QoS to the video service in scenarios comprising legacy DCF-based and IEEE 802.11e stations. This is due to the fact that, under theses schemes, the video packets have always to wait for a minimum of eight backoff slots in order to comply with the highest priority assigned to the voice traffic. Under these schemes, the presence of voice and DCF stations may even result in starvation to the video flows. Moreover, the implementation of these mechanisms implies that the stations have to monitor the channel conditions in order to dynamically tune up the actual values of the key system parameters, such as the threshold and window size.

Taking into account these observations, in the next section, we propose a new IEEE 802.11e based QoS mechanism capable of providing QoS support to the video service even in the presence of legacy IEEE 802.11 (DCF) based stations.

4 B-EDCA: A New IEEE 802.11e Based QoS Mechanism

Due to the fact that the IEEE 802.11e interface cards will take over the WLAN market, replacing the use of legacy IEEE 802.11 interface cards in most WLAN applications, an important number of networking scenarios will consist of a hybrid configuration comprising legacy IEEE 802.11-based stations and IEEE 802.11e-based stations. Under these scenarios, EDCA, RT-FCR and AFEDCF perform poorly, especially they are unable to provide the QoS required by the video traffic.

Based on limitations of these mechanisms, we propose a new IEEE-802.11e based QoS mechanism compatible with the IEEE 802.11e specifications and capable of providing QoS support, particularly to video applications.

Bearing in mind that the DCF and EDCA mechanisms may have to interwork, the standard committee has set up the system parameters given in Table I. These values have been set up in order to ensure compatibility between both services and that the EDCA mechanism has to be able to provide QoS guarantees to time-constrained applications, namely voice and video traffic. As shown in Table I, the EDCA mechanism makes the use of a smaller contention window for the voice and video applications.

Based on the results obtained in one of our previous studies [10], we have found out that the IFS (denoted AIFS in the EDCA draft standard) is the most important and critical parameter enabling the provisioning of QoS to multimedia applications. This is particular true when a large number of stations attempt to gain access to the channel, since under these conditions, the stations will often have to stop decrementing their backoff counters. Recall that every time that a station stops decrementing its counter, the station must wait an AIFS before resuming the count down.

Table 1. Parameter settings specified in standards [1], [2]

	AC	IFS	CW_{min}	CW_{max}
DCF	-	2 x Slot_time + SIFS	31	1023
EDCA	Vo	2 x Slot_time + SIFS	7	15
	Vi	2 x Slot_time + SIFS	15	31
	Be	3 x Slot_time + SIFS	31	1023
	Bk	7 x Slot_time + SIFS	31	1023

One possible solution will be to set up AIFS=1 for the voice and video applications. In this way, they will increase their chances to gain access to the channel. However, setting up AIFS=1 to these two services is incompatible with the HCCA. As already explained, the HC should be able to take the control of the channel at any time. This is to say, the HCCA should hold the highest priority over all the services to be supported by the standard.

In order to introduce our proposal, we take a closer look at the mode of operation of the DCF and EDCA schemes, and particularly on the role played by the IFS (AIFS) parameter. The IFS (AIFS) is used in the following two cases:

1. In the **Idle** state. when the station becomes active has to sense the channel during an interval whose length is determined by IFS: If the channel is sensed free, the station can initiate the packet transmission. Otherwise, the station executes the backoff algorithm.
2. In every transfer from the **Defer** state to the **Backoff** state. In other words, every time after having sensed the channel free during an interval of length IFS.

According to the current DCF and EDCA standards, the same values for the IFS parameter should be used regardless of the state in which the station is (see Table 1). Based on the previous observation, we then propose to use a different set of IFS values depending on the state in which the station is. We have however to ensure not to compromise the operation of the HCF, and in particular to ensure that it holds at all times the highest priority. We then propose the following parameter setting:

1. In the **Idle** state. The stations will use the IFS values as specified in the IEEE 802.11e draft standard (see Table I) including the Hybrid Coordination Function. This also ensures compatibility with the IEEE 802.11 (DCF) mechanism.

2. In every transfer from the **Defer** state to the **Backoff** state, we propose to use a different parameter, equivalent to the IFS, denoted from now on by BIFS. We then propose setting up this parameter to one, i.e., BIFS = 1, for the voice and video services. In this way, we improve considerably the performance of voice and video applications, increasing their priorities with respect to other flows (included the traffic generated by DCF-based stations). This setting also ensures that the HC will keep the highest priority. According to this mechanism, the stations must wait at least one additional slot during the backoff procedure before being allowed to transmit since the backoff interval is set within the [1, CW+1] range. In turn, the HC is allowed to take the control at the end of the IFS. To improve further the provisioning of QoS guarantees to the time-constrained applications when the network is highly loaded, we propose increasing the assigned value to BIFS used by the Best-Effort traffic, with respect to the specified in [2]. We then propose using the set of values for BIFS to 1-1-4-7 for voice, video, best-effort and background traffics, respectively.

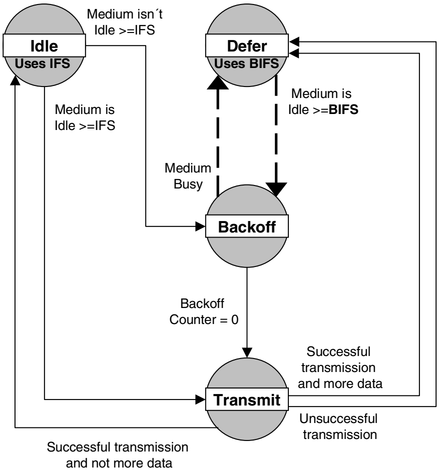


Fig. 1. B-EDCA Proposed Mechanism

In Figure 1, we have explicitly indicated the instances where the BIFS parameter should be used. This is essentially the major change with respect to the current EDCA standard. Our proposal essentially reduces to the minimum acceptable value, the waiting time required to continue decrementing the backoff counter used by the time-constrained applications. This minimum value is fully compatible with the operation modes of the DCF and HCCA functions.

5 Performance Evaluation

In this section, we carry out a performance analysis of our proposed mechanism. We show that the performance of EDCA can be considerably improved by using the compatible B-EDCA mechanism. In this part of our study, we compare the performance of our proposed scheme with the EDCA, RT-FCR, AFEDCF mechanisms by considering a scenario of a wireless LANs comprising IEEE 802.11-based stations and stations supporting one of the QoS-aware mechanisms under study. Throughout our study, we have made use of the OPNET Modeler tool 10.0 [11].

5.1 Scenario

In our simulations, we model an IEEE 802.11b wireless LAN cell comprising legacy DCF-based stations and stations implementing one of the four QoS-aware mechanism stations under consideration. The QoS-aware mechanism based stations support four different types of services: voice (Vo), video (Vi), best-effort (BE) and background (BK). This classification is in line with the IEEE802.1D standard specifications. The DCF based stations support data traffic. We assume the use of a wireless LAN consisting of several wireless stations and an access point connected to a wired node that serves as sink for the flows from the wireless domain. All the stations are located within a *Basic Service Set* (BSS), i.e., every station is able to detect the transmission from any other station.

Each wireless station operates at 11 Mbit/s IEEE 802.11b mode and transmits a single traffic type to the access point. We assume the use of constant bit-rate voice sources encoded at a rate of 16 kbits/s according to the G.728 standard [12]. The voice packet size has been set to 168 bytes including the RTP/UDP/IP headers. For the video applications, we have made use of the traces generated from a variable bit-rate H.264 video encoder [13]. We have used the sequence mobile calendar encoded on CIF format at a video frame rate of 25 frames/sec. The average video transmission rate is around 480 kbits/s with a packet size equal to 1064 bytes (including RTP/UDP/IP headers). The best-effort, background and DCF traffics have been created using a *Pareto* distribution traffic model. The average sending rate of best-effort and background traffic is 128 kbit/s, using a 552 bytes packet size (including TCP/IP headers). The average sending rate of DCF traffic is 256 kbit/s, using a 552 bytes packet size (including TCP/IP headers). All traffic sources are randomly activated within of the interval [1,1.5] seconds from the start of the simulation. We have simulated two minutes of operation for each given scenario.

For all the scenarios, we have assumed that one fifth of the stations support one of the five kinds of services: voice, video, BE, BK and DCF applications. We start by simulating a WLAN consisting of five wireless stations (each one supporting a different type of traffic). We then gradually increase the *Total Offered Load* of the wireless LAN by increasing the number of stations by five. In this way, the stations are always incorporated into the system in a ratio of 1:1:1:1:1 for voice, video, BE, BK and DCF, respectively. We increase the number of stations 5 by 5 starting from 5 and up to 40. In this way, the normalized offered load is increased from 0.14 up to 1.12. By exceeding the channel capacity, we should be able to evaluate the effectiveness of the QoS-aware mechanisms on guaranteeing the QoS required by the time-constrained applications. When choosing the parameter settings to use for the DCF and EDCA mechanisms under study, we have used the settings recommended by the standards [1], [2] (see Table I). The parameter settings used for the RT-FCR and AFEDCF mechanisms under study have been taken from references [7] and [9], respectively. The parameter settings for the B-EDCA mechanism have been defined by following the guidelines provided in Section 4.

For the purpose of our performance study, the four metrics of interest are: throughput, media access delay, delay distribution and packet loss rate. To be able to compare the results at different loads (traffic patterns of different applications), we have preferred plotting the normalized throughput rather than the absolute throughput. The normalized throughput is calculated as the percentage of the offered load actually delivered to destination. In order to limit the delay experienced by the video and voice applications, the maximum time that video packet and voice packet may remain in the transmission buffer has been set to 100*ms* and 10*ms*, respectively. These time limits are in line with the values specified by the standards and in the literature. Whenever a video or voice packet exceeds these upper bounds, it is dropped. The loss rate due to this mechanism is given by the *packet loss rate due to deadline*. Our measurements started after a warm-up period allowing us to collect the statistics under steady-state conditions. Each point in our plots is an average over thirty simulation runs, and the error bars indicate the 95% confidence interval.

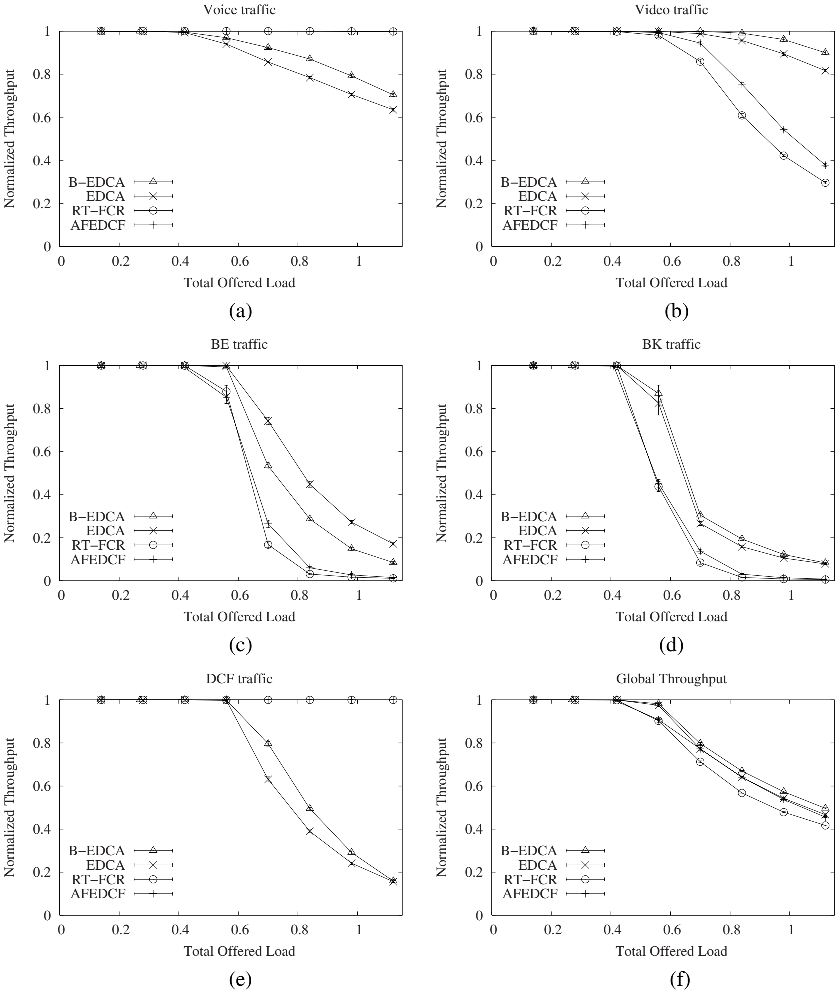
5.2 Results

Figure 2 shows the normalized throughput obtained for the Vo, Vi, BE and BK services when making use of each one of the four mechanisms being considered. Figure 2a shows that B-EDCA mechanism outperforms the EDCA mechanism in providing a better service to the voice traffic. This shows that by reducing the effectiveness of setting the BIFS parameter to one. The figure also shows that the RT-FCR and AFEDCF mechanisms obtain the best results for the voice traffic. This is due to the fact that, under theses schemes, the highest priority is given to the voice traffic, all the other traffic types have to wait a minimum of eight backoff slots. However, under these schemes, the presence of voice and DCF stations produces starvation in the video flows, see Figure 2b. For video traffic, under RT-FCR and AFEDCF mechanisms, when the load exceeds 0.5, the throughput of the video traffic quickly decreases. The decrease on the video throughput is mainly due to the fact that under the RT-FCR and AFEDCF mechanisms, the DCF based stations have a higher priority than the one given to the video stations, see Figure 2e. Figure 2b also shows that B-EDCA obtain the best results for the video traffic. Again, for the case of the video traffic, the B-EDCA mechanism outperforms the EDCA mechanism. In the case of the BE and BK traffics (figures 2c and 2d), these are severely affected as the network load is increased. Figure 2f shows the overall throughput for all the services under study. It is clear that the B-EDCA exhibits the highest normalized throughput. This is due to the reduction of the collision rate with respect to EDCA mechanism, and to the fact that in the RT-FCR and AFEDCF mechanisms, all the flows (except voice) must wait eight additional backoff slots.

These phenomena also explain the access delay performance. Figure 3 shows the mean access delay per voice and video service classes. Figure 3a shows that the B-EDCA reduces up to 50% the mean access delay experienced by the voice traffic when using the EDCA mechanism. Figure 4b shows that the B-EDCA scheme exhibits the best results for the video service. It can also be observed that the mean access delays for RT-FCR and AFEDCF mechanisms are very close to the video deadlines; this in turn translates in a high packet loss rate (Figure 5).

Figure 4 shows the cumulative distribution function of the access delay for all mechanism operating at a load close to 0.80. Figure 4a shows that B-EDCA mechanism outperforms the EDCA mechanism for the voice traffic. Figure 4b also shows that B-EDCA obtain the best results for the video traffic.

Figures 5a and 5b depict the packet loss rate due to the missing of the transmission deadline for the voice and video traffic services, respectively. The B-EDCA scheme provides the best results for the video traffic. The B-EDCA scheme is able to ensure the proper transmission of the video traffic even at loads as high as 0.8.



(a) (b) (c) (d) (e) (f)

Fig. 2. Average Normalized Throughput: a) Voice, b) Video, c) Best-Effort d) Background e) DCF Traffic and f) Total Traffic

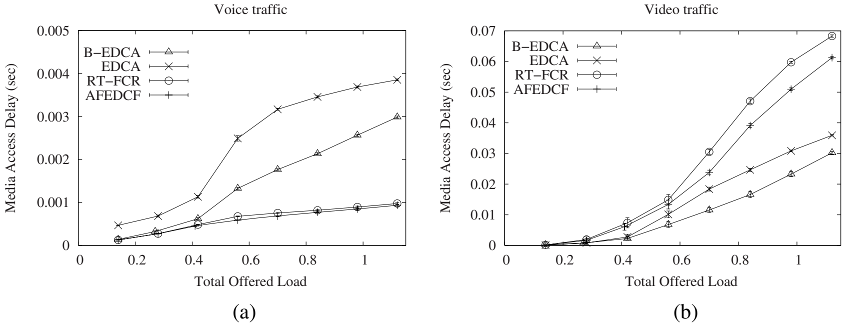


Fig. 3. Average Access Delay: a) voice, b) video

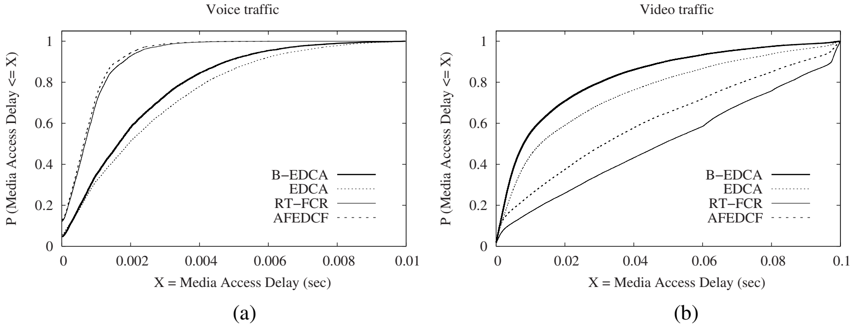


Fig. 4. Cumulative Distribution (CDF) of the Access Delays: a) voice, b) video

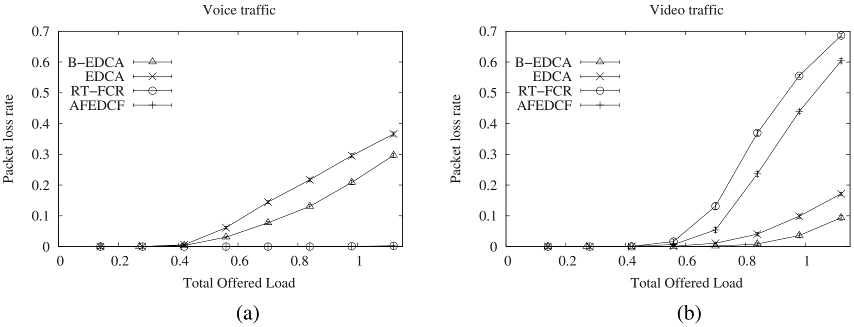


Fig. 5. Packet Loss Rate due to Deadline: a) voice and b) video

6 Conclusions

In this paper, we have proposed a new IEEE 802.11e based QoS protocol design capable of providing QoS support in environments where legacy DCF based stations may also be present. Our proposal has been based in using the minimum waiting time necessary to continue decrementing the backoff counter of the multimedia flows. Furthermore, our proposal complies with the HCF operation proposed by the IEEE 802.11e standards. Our results obtained have shown that B-EDCA mechanism outperforms the EDCA mechanism and two other relevant mechanisms reported in the literature.

References

1. LAN MAN Standards Committee of the IEEE Computer Society, ANSI/IEEE Std 802.11, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, 1999 Edition.
2. IEEE 802 Committee of the IEEE Computer Society, IEEE P802.11e/D13.0 Draft Amendment to IEEE Std 802.11, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements”, April 2005.
3. F. Mico, P. Cuenca and L.Orozco Barbosa “QoS Mechanisms for IEEE 802.11 Wireless LANs”. Lecture Notes in Computer Science. Vol. 3079. pp. 609-623, 2004.
4. J. Villalón, P. Cuenca, L. Orozco-Barbosa. “QoS Provisioning Mechanisms for IEEE 802.11 WLAN: A Performance Evaluation”. Proceedings of 10th IFIP International Conference on Personal Wireless Communications. Colmar, August 2005.
5. A. Lindgren, A. Almquist and O. Schelén, “Quality of Service Schemes for IEEE 802.11 Wireless LANs - An Evaluation”, Journal of Special Topics in Mobile Networking and Applications (MONET), Vol. 8, No. 3, pp. 223-235, June 2003.
6. W. Pattara-Atikom, P. Krishnamurthy and S. Banerjee, “Distributed Mechanisms for Quality of Service in Wireless LANs”, IEEE Wireless Communications, Vol. 10, No. 3, pp. 26-34, June 2003.
7. Y. Kwon, Y. Fang and H. Latchman, “Design of MAC Protocols with Fast Collision Resolution for Wireless Local Area Networks”. IEEE Transactions on Wireless Communications, Vol. 3, No.3. pp. 793-807, May 2004.
8. L. Romdhani, Q. Ni and T. Turletti, “Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks”, in Proceedings of IEEE WCNC, New Orleans, pp. 1373-1378. March 2003.
9. M. Malli, Q. Ni, T. Turletti and C. Barakat “Adaptive Fair Channel Allocation for QoS Enhancement in IEEE 802.11 Wireless LANs”, Proceedings of IEEE ICC, Paris, June 2004.
10. J. Villalón, P. Cuenca, L. Orozco-Barbosa, “On the Effectiveness of IEEE 802.11e QoS Support in Wireless LAN: A Performance Analysis”. Lecture Notes in Computer Science. Vol. 3726. pp. 605-616, 2005.
11. Opnet.Technologies.Inc. OPNET Modeler 10.0, 1987-2004. http://www.opnet.com
12. ITU-T Recommendation G.728, “Coding of Speech at 16 kbit/s using Low-Delay Code Excited Linear Prediction”, Std., September 1992.
13. ITU-T Recommendation H.264, “Advanced Video Coding For Generic Audiovisual Services”. May 2003.

A Lagrangian Approach for the Optimal Placement of Wireless Relay Nodes in Wireless Local Area Networks

Aaron So and Ben Liang

Department of Electrical and Computer Engineering,
University of Toronto, Toronto, Ontario, Canada M5S 3G4
{aaronso, liang}@comm.utoronto.ca

Abstract. The throughput capacity of WLANs can be improved by a carefully designed relay infrastructure. In this work, we propose an optimization formulation based on Lagrangian relaxation and a subgradient algorithm to compute the best placement of a fixed number of relay nodes (RNs) in a WLAN. We apply this optimization framework to a multi-rate WLAN based on the IEEE 802.11g standard under Rayleigh fading. We then study the expected throughput capacity of a WLAN with relay infrastructure and investigate how the optimal placement of RNs is affected by the number of RNs, path-loss characteristics, and the traffic pattern. Our numerical results show that, in some network scenarios, more than 120% performance gain can be achieved when RNs are strategically installed in the network. Furthermore, we also show that for a wide range of system parameters, optimally placed RNs can significantly increase the network throughput capacity over random placement.

Keywords: WLAN, immobile relays, throughput capacity, optimal placement.

1 Introduction

Wireless Local Area Networks (WLANs), which provide low-cost wireless broadband data access for mobile Internet users, are expected to create a plethora of business opportunities. Currently, the most commonly implemented WLANs in North America are based on the IEEE 802.11b/g standards, which are capable of supporting bit rates up to 11Mbps and 54Mbps respectively in the 2.4GHz spectrum. As the number of hotspot users proliferates and the demand from wireless Internet users increases, new strategies have to be employed to increase the throughput of future WLANs.

The multi-rate capability of modern WLAN equipments and a relay infrastructure can work synergically to improve the throughput capacity of a WLAN. In a WLAN with relay infrastructure, the source can either transmit its data to the destination directly, or relay its data via a relay node (RN). If a circuitous route can result in a higher bit rate than the direct route, the source should use the relay node to relay its data.

In this study, we investigate the optimal placement of RNs such that the throughput capacity of a WLAN can be maximized. Toward this end, we propose an analysis and optimization framework that exploits the multi-rate capability of the WLAN physical layer. Our main contributions are the following:

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 160–172, 2006.

- Present a tractable discretized re-formulation of the problem of optimal RN placement, which allow us to restrict the locations of the RNs.
- Solve the discrete version of the problem by computing an upper bound and a lower bound of the solution that converge toward each other, through Lagrangian relaxation and subgradient algorithm.
- Investigate the expected throughput capacity of a WLAN with relay infrastructure and how the optimal placement of RNs is affected by the number of RNs, path-loss characteristics, and traffic pattern.

The rest of this paper is organized as follows. In Section 2, we review the related work in multihop wireless networks. In Section 3, we describe the relaying architecture and define the network throughput capacity. In Section 4, we cast the general RN placement optimization problem and re-formulate it to a tractable discrete problem. We then show how this problem can be solved by Lagrangian relaxation and a subgradient algorithm. In Section 5, we present a model for IEEE 802.11g multi-rate WLAN under Rayleigh fading. In Section 6, we discuss the convergence time of the proposed optimization algorithm and show effect of different system parameters on the strategic placement of the RNs. Finally, concluding remarks are given in Section 7.

2 Related Work

There has been much research in relaying and routing through mobile nodes. Inspired by recent advances in ad hoc networking [1], the concept of using peer mobile hosts to relay data has been explored in the context of cellular networks [2]. In a more recent work, the problem of joint routing, link scheduling and power control in such multihop networks has been investigated [3]. Moreover, issues about frequency assignment and frequency recycling in such multihop networks have been addressed in [4]. In the context of multi-rate WLAN, [5] and [6] have shown that by using other mobile hosts to perform relaying, the performance of the network can be improved under DCF and PCF respectively.

However, the concept of using immobile relay nodes to relay traffic, which is what we consider in this paper, has received less attention. The *iCar* architecture [7] is one such example for the cellular environment. Immobile relay nodes have several advantages when compared with mobile relay nodes. First, because of their sedentariness, it is reasonable to assume that they have access to power supply. Consequently, energy is not a constraint. Second, the fixed relay nodes can be optimally configured to maximize their beneficial effects. The problem of fixed relay placement to maximize WLAN capacity was first studied in [8]. In [9], we proposed an efficient extension point placement algorithm aiming at improving the network layer throughput of a rectilineal network, using a divide-and-conquer searching algorithm. In this work, we explore the utilization of relay infrastructure in a discrete multi-rate WLAN, and analytically derive the optimal placement of relay nodes in such WLANs in a general network environment.

3 Relaying Architecture and Design Objectives

The system under consideration is analogous to a Basic Service Set (BSS) of an IEEE 802.11 WLAN. In this network configuration, there is an AP which is connected to the wired network, and this access point provides wireless coverage to a local area. In a network with no RN, MHs located within this coverage area directly communicate with this AP. In a network with RNs, mobile hosts located at different locations are associated with either the AP or a suitable RN. If an MH is associated with an RN, this MH will treat the selected RN as an AP and only communicate with the RN, so that all packets between the AP and the MH are relayed by the RN.

The AP communicates with each MH (possibly through an RN) in its coverage area in a round robin fashion, thus dividing the time axis into time-varying *packet transaction cycles*. In each cycle, the AP transmits a downlink packet to the chosen MH, and the MH transmits a uplink packet to the AP. In this study, we assume the lengths of the uplink and downlink packets may be unequal but are fixed, and the AP always has a packet to send to each active MH and vice versa.

We assume the transmission schedule for all transmitters is decided perfectly by the AP, and model the system as a single-channel fully-connected network. In other words, at any given time, only one transmitter is allowed to transmit, so that no packet collision is experienced at a receiver.

Let x be the total number of bits of an uplink and a downlink packet combined. Let T_i represents the packet transaction time of an AP-MH pair in the i^{th} cycle. By the Law of Large Numbers, the throughput capacity of the network is defined as

$$C = \lim_{n \to \infty} \frac{nx}{\sum_{i=1}^{n} T_i} = \frac{x}{E[T_i]} \quad . \tag{1}$$

Therefore, in order to maximize the throughput capacity of the network, we need to minimize $E[T_i]$. Thus, the design objective of our system is to minimize the expected time that an AP-MH pair completes a single downlink-uplink packet exchange, which we call the *packet transaction time* in this paper.

We further define the *packet transmission time*, $T(l, P, x)$, as the expected time for a transmitter to send an x-bit packet to a receiver, where l is the distance between the transmitter and receiver, P is the reference power of the transmitter, and x is the size of the packet to be transmitted. Next, we first propose an optimization method for the placement of RNs for a generic function $T(l, P, x)$, which can be obtained from theoretical models or by regression models based on site-survey results. In Section 5, we present a case study for $T(l, P, x)$ based on the IEEE 802.11g physical layer specifications with large-scale propagation path loss and Rayleigh fading.

4 Relay Node Placement Optimization

In this section, we first provide an analytical framework to derive the expected packet transaction time with respect to different RN placements. This analytical framework is then used to cast the RN placement problem as an optimization problem. We show that the resulting optimization can be converted to a form similar to the *p*-median problem [10] with an additional constraint. Finally, we present an efficient solution based on Lagrangian relaxation and a subgradient optimization algorithm.

4.1 RN Relaying

In the elemental scenario, we consider a single MH and one RN, on a plane where the AP is at the origin, as shown in Fig. 1. If the MH does not use the RN, the expected packet transaction time is

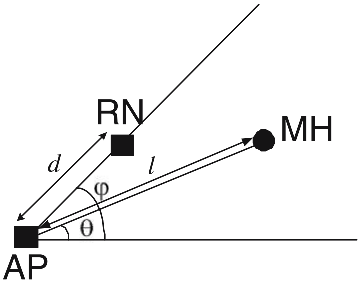
$$T_{norn}(l) = T(l, P_a, x_d) + T(l, P_m, x_u). \tag{2}$$

If the MH uses the RN to relay its packet, the expected packet transaction time is

$$T_{rn}(l, d, \theta, \varphi) = T(d, P_a, x_d) + T(\eta, P_r, x_d) + T(d, P_r, x_u) + T(\eta, P_m, x_u), \tag{3}$$

where $\eta = \sqrt{l^2 + d^2 - 2ld \cos|\theta - \varphi|}$.

By using an RN, the same data packet has to be transmitted twice. As a result, the RN may or may not be beneficial to an MH. An MH will use an RN to facilitate its communication with the AP only if such usage result in a smaller expected packet transaction time, i.e., if $T_{rn}(l, d, \theta, \varphi) < T_{norn}(l)$.



- x_d = downlink packet length (bits).
- x_u = uplink packet length (bits).
- $x = x_d + x_u$.
- $\beta = \frac{x_d}{x_d + x_u}$ = downlink proportion.
- P_a = reference power of AP.
- P_r = reference power of RN.
- P_m = reference power of MH.

Fig. 1. Single user scenario with one relay node

4.2 Throughput Capacity Maximization with Multiple RNs

We assume that a fixed number, N, where $N > 1$, of relay nodes are available for an AP in the WLAN system. Fig. 2 shows a simple example, where an AP serves the outdoor area of a campus. There are 3 RNs available and they are placed around the AP. The coverage area, which can be in any shape, is fitted inside a circle with radius L and centered at the AP. A vector, $\underline{d}$, is used to represent the displacement of RNs with respect to the AP, and a vector $\underline{\varphi}$ is used to represent the angle between a predefined reference base line with respect to the radial line which each RN resides. Thus, we have

$$\underline{d} = [d_1, d_2, ..., d_N]^T, \qquad \underline{\varphi} = [\varphi_1, \varphi_2, ..., \varphi_N]^T, \tag{4}$$

where $0 < d_i \leq L, \forall i$ and $0 \leq \varphi_1 \leq \varphi_2 \leq \cdots \leq \varphi_N \leq 2\pi$. Moreover, the locations of RNs may be restricted due to the geographical topology. Let the set of locations where an RN can be installed be S. For example, in Fig. 2, S may be a subset of the perimeters of the buildings.

The MHs are distributed in the coverage area of the network with the probability density function $f(l, \theta)$. An MH may either communicate with the AP directly or select

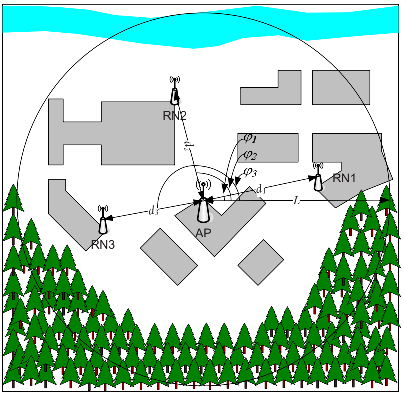


Fig. 2. Multi-user two-dimensional WLAN with multiple RNs

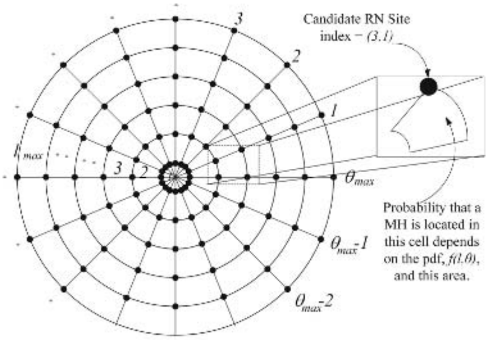

Candidate RN Site index – (3,1)
3
2
1
l_{max}
3
2
θ_{max}
θ_{max}-1
θ_{max}-2
Probability that a MH is located in this cell depends on the pdf, f(l,θ), and this area.

Fig. 3. Discretization of the network

the most suitable RN. Therefore, for a particular RN placement, $(d_i, \varphi_i) \in S$ for $i = 1, ..., N$, the expected packet transaction time of the network can be computed as

$$\overline{T_{rn}(\underline{d}, \underline{\varphi})} = \int_0^{2\pi} \int_{0+\epsilon}^{L} l f(l,\theta) \min \left[T_{norn}(l), \min_{1 \le k \le N} T_{rn}(l, d_k, \theta, \varphi_k) \right] dl d\theta, \quad (5)$$

where $\epsilon > 0$ is small. Using (5), we have the following optimization problem:

$$\begin{aligned} \text{Objective:} \quad & \min_{\underline{d},\underline{\varphi}} \ \overline{T_{rn}(\underline{d}, \underline{\varphi})} \\ s.t. \quad & 0 < d_i \le L, \qquad \forall i \\ & 0 \le \varphi_1 \le \varphi_2 \le \cdots \le \varphi_N \le 2\pi \\ & (d_i, \varphi_i) \in S \qquad \forall i. \end{aligned} \quad (6)$$

Clearly, this problem is difficult to solve directly. However, in reality, there is no need to determine the RN placement with sub-meter granularity. Hence, we can calculate the approximate value of (5) by discretizing the network into a large but finite number of areas, where a mobile host is located at each area with a certain probability. Then, the integral in (5) can be interpreted as a Riemann sum.

4.3 Problem Reformulation

As shown in Fig. 3, we can divide the entire network into θ_{max} equal-size sectors, and each sector is then divided into l_{max} equal-length cells.[1] A mobile host is located at the corner of each cell with a certain probability, and the area of the cell represents the area that this mobile host occupies. Moreover, the corner of each cell also represents a candidate site of the RN set. Each MH or RN candidate site can be uniquely identified by its radial line number and its cell number. For example, the selected site in Fig. 3 lies on the third cell of the first radial line. Thus, this site is indexed by (3,1). For notation

[1] In our numerical analysis, the network is divided into approximate 100 thousand cells.

purpose, we use (i,j) to describe the location of an MH, while we use (δ,τ) to represent the location of an RN candidate site. We define the following notations:

$$\begin{aligned}
\Delta\theta &= \frac{2\pi}{\theta_{max}}, \quad \Delta l = \frac{L}{l_{max}}, \quad \underline{\delta} = (\delta_1,...,\delta_N)^T, \quad \underline{\tau} = (\tau_1,...,\tau_N)^T,\\
d_k &\approx \delta_k \Delta l, \quad \text{for } 1 \le k \le N, \text{ and } \delta_k \in \mathbf{Z}^+, \text{ and } 1 \le \delta_k \le l_{max},\\
\varphi_k &\approx \tau_k \Delta\theta, \quad \text{for } 1 \le k \le N, \text{ and } \tau_k \in \mathbf{Z}^+, \text{ and } 1 \le \tau_k \le \theta_{max},\\
h_{(a,b)} &= \int_{(b-1)\Delta\theta}^{b\Delta\theta} \int_{(a-1)\Delta l}^{a\Delta l} lf(l,\theta)dld\theta, \;\; a \in [1, l_{max}], b \in [1, \theta_{max}],\\
S' &= M(S),
\end{aligned}$$

where $M(\cdot)$ is the mapping of a set from the continuous space to the discrete space. Furthermore, for notation simplification, we let $T_{rn}^{\Delta l,\Delta\theta}(i,\delta,j,\tau) = T_{rn}(i\Delta l, \delta\Delta l, j\Delta\theta, \tau\Delta\theta)$, and $T_{norn}^{\Delta l}(i) = T_{norn}(i\Delta l)$. Then, (5) can be approximated as

$$\begin{aligned}
\overline{T_{rn}(\underline{d},\underline{\varphi})} &= \int_0^{2\pi} \int_{0+\epsilon}^{L} lf(l,\theta) \min\left[T_{norn}(l), \min_{1\le k\le N} T_{rn}(l,d_k,\theta,\varphi_k)\right] dld\theta\\
&\approx \sum_{i=1}^{l_{max}} \sum_{j=1}^{\theta_{max}} \min\left[T_{norn}^{\Delta l}(i), \min_{1\le k\le N} T_{rn}^{\Delta l,\Delta\theta}(i,\delta_k,j,\tau_k)\right] h_{(i,j)}\\
&= \overline{T_{rn}(\underline{\delta},\underline{\tau})}
\end{aligned} \tag{7}$$

To facilitate the minimization of $\overline{T_{rn}(\underline{\delta},\underline{\tau})}$, we define two sets of decision variables, $\mathbf{X}$ and $\mathbf{Y}$. $X_{(\delta,\tau)} = 1$ if an RN is placed in position (δ,τ); otherwise, $X_{(\delta,\tau)} = 0$. Moreover, $Y_{(i,j),(\delta,\tau)} = 1$ if the MH (i,j) is served by RN (δ,τ); otherwise, $Y_{(i,j),(\delta,\tau)} = 0$.

Note that $X_{(0,0)} = 1$ because the access point is always present. Moreover, since $X_{(0,0)} = 1$, we have $Y_{(i,j),(0,0)} = 1$ only if the MH at (i,j) is served directly by the AP. Thus, (6) can be reformulated as

$$\min_{\mathbf{X},\mathbf{Y}}: \sum_{i=1}^{l_{max}} \sum_{j=1}^{\theta_{max}} \left[h_{(i,j)} T_{norn}^{\Delta l}(i) Y_{(i,j),(0,0)} + \sum_{\delta=1}^{l_{max}} \sum_{\tau=1}^{\theta_{max}} h_{(i,j)} T_{rn}^{\Delta l,\Delta\theta}(i,\delta,j,\tau) Y_{(i,j),(\delta,\tau)}\right] \tag{8}$$

$$s.t. \;\; Y_{(i,j),(0,0)} + \sum_{\delta=1}^{l_{max}} \sum_{\tau=1}^{\theta_{max}} Y_{(i,j),(\delta,\tau)} = 1 \quad \forall \;\; (i,j) \tag{9}$$

$$\sum_{\delta=1}^{l_{max}} \sum_{\tau=1}^{\theta_{max}} X_{(\delta,\tau)} = N \tag{10}$$

$$X_{(0,0)} = 1 \tag{11}$$

$$Y_{(i,j),(\delta,\tau)} - X_{(\delta,\tau)} \le 0 \qquad \forall \;\; (i,j),(\delta,\tau) \tag{12}$$

$$X_{(\delta,\tau)} = 0 \qquad \forall \;\; (\delta,\tau) \notin S' \tag{13}$$

Objective (8) minimizes the expected minimum packet transaction time of all MHs in the network. Constraint (9) requires each MH to be assigned to exactly one RN or the AP. Constraint (10) states that exactly N RNs are to be located. Constraint (11) states that the AP is always present. Constraint (12) requires that the MH at (i, j) can be assigned to an RN at (δ, τ) only if an RN is installed at location (δ, τ). Constraint (13) required that the RNs are not located in the infeasible sites.

4.4 An Optimization-Based Lagrangian Relaxation Iterative Algorithm

Lagrangian relaxation with subgradient optimization can be used to provide approximate solution to many NP-hard problems efficiently. Thus, noting the distinctive characteristics of our RN placement formulation, we propose to solve our optimization problem in (8) as follows.

Step 1: Setting up. We relax constraint (9) and obtain (14), where $\lambda_{(i,j)}$ are the Lagrange multipliers. In our numerical analysis below, all $\lambda_{(i,j)}$ values are initialized to 5000.

$$\begin{aligned}
\max_{\lambda}\min_{\mathbf{X},\mathbf{Y}} \quad & \sum_{i=1}^{l_{max}}\sum_{j=1}^{\theta_{max}} \lambda_{(i,j)} \left[1 - Y_{(i,j),(0,0)} - \sum_{\delta=1}^{l_{max}}\sum_{\tau=1}^{\theta_{max}} Y_{(i,j),(\delta,\tau)}\right] + \\
& \sum_{i=1}^{l_{max}}\sum_{j=1}^{\theta_{max}} \left[h_{(i,j)} T_{norn}^{\Delta l}(i) Y_{(i,j),(0,0)} + \sum_{\delta=1}^{l_{max}}\sum_{\tau=1}^{\theta_{max}} h_{(i,j)} T_{rn}^{\Delta l,\Delta\theta}(i,\delta,j,\tau) Y_{(i,j),(\delta,\tau)}\right] \\
= & \sum_{i=1}^{l_{max}}\sum_{j=1}^{\theta_{max}} \lambda_{(i,j)} + \sum_{i=1}^{l_{max}}\sum_{j=1}^{\theta_{max}} \left[(h_{(i,j)} T_{norn}^{\Delta l}(i) - \lambda_{(i,j)}) Y_{(i,j),(0,0)} + \right. \\
& \left. \sum_{\delta=1}^{l_{max}}\sum_{\tau=1}^{\theta_{max}} (h_{(i,j)} T_{rn}^{\Delta l,\Delta\theta}(i,\delta,j,\tau) - \lambda_{(i,j)}) Y_{(i,j),(\delta,\tau)} \right] \qquad (14) \\
s.t. \quad & (10),\ (11),\ (12),\ (13) \text{ are satisfied .}
\end{aligned}$$

Step 2: Solving the simplified problem. For fixed values of the Lagrange multipliers, we want to minimize the objective function (14). Since the values of $\lambda_{(i,j)}$ are fixed, the first term in the objective function, which is just the sum of all Lagrangian multipliers, is a constant. To minimize (14), we begin by computing the value of

$$V_{(\delta,\tau)} = \sum_{i=1}^{l_{max}}\sum_{j=1}^{\theta_{max}} \min(0, [h_{(i,j)} T_{rn}^{\Delta l,\Delta\theta}(i,\delta,j,\tau) - \lambda_{(i,j)}]) \quad \forall(\delta,\tau) \in S' \qquad (15)$$

We then find the N smallest values of $V_{(\delta,\tau)}$ and set the corresponding values of $X_{(\delta,\tau)} = 1$ and all other values of $X_{(\delta,\tau)} = 0$. We then set$Y_{(i,j)(\delta,\tau)} = 1$ if $h_{(i,j)} T_{rn}^{\Delta l,\Delta\theta}(i,\delta,j,\tau) - \lambda_{(i,j)} < 0$ and $X_{(\delta,\tau)} = 1$. Moreover, since $X_{(0,0)} = 1$, we set $Y_{(i,j)(0,0)} = 1$ if $h_{(i,j)} T_{norn}^{\Delta l}(i) - \lambda_{(i,j)} < 0$. All other Y's are set to zero.

Step 3: Updating the lower and upper bounds. For each iteration of this process, an upper bound and lower bound of the original objective function (8) need to be determined. From Step 2, N RN candidate sites are selected. The expected minimum packet transaction time with this particular placement can be calculated by using (7). This value is an upper bound of (8). The lower bound for the current iteration is simply the objective function (14) with the values of $\mathbf{X}$ and $\mathbf{Y}$ found in Step 2 [11].

Step 4: Modifying the Lagrange multipliers. The Lagrange multipliers are revised using a standard subgradient optimization procedure [11]. At the n^{th} iteration of the Lagrangian procedure, we first compute the step size by

$$t^n = \frac{A^n(UB - \mathcal{L}^n)}{\sum_{i=1}^{l_{max}} \sum_{j=1}^{\theta_{max}} \left[Y^n_{(i,j),(0,0)} + \sum_{\delta=1}^{l_{max}} \sum_{\tau=1}^{\theta_{max}} Y^n_{(i,j),(\delta,\tau)} - 1 \right]^2}, \tag{16}$$

where UB and $\mathcal{L}^n$ are the upper and lower bounds found from Step 3, $Y^n_{(i,j),(\delta,\tau)}$ is the optimal value of $Y_{(i,j),(\delta,\tau)}$ at the n^{th} iteration, and A^n is a constant updated as follows. We begin with $A^1 \leq 2$ an arbitrary small positive number. At each iteration, the value of A^n is halved if $\mathcal{L}^n$ has not increased in c_A consecutive iterations. In our numerical analysis, we use $A^1 = 2$ and $c_A = 4$. Then, the Lagrangian multipliers are updated by

$$\lambda^{n+1}_{(i,j)} = \max \left[0, \lambda^n_{(i,j)} - t^n \left(Y^n_{(i,j),(0,0)} + \sum_{\delta=1}^{l_{max}} \sum_{\tau=1}^{\theta_{max}} Y^n_{(i,j),(\delta,\tau)} - 1 \right) \right]. \tag{17}$$

Step 5: Iteration and termination. The algorithm terminates when any one of the following conditions is true:

1. A predefined number of iterations are completed.
2. The upper bound equals or is close enough to the lower bound.
3. A^n is small, such that the changes in $\lambda_{(\delta,\tau)}$ becomes too small. Such small changes are not likely to help solve the problem.

Otherwise, we repeat from Step 2.

5 IEEE 802.11 Model and Packet Transmission Time

In this section, as a sample case study, we derive the expected packet transmission time based on the IEEE 802.11g bit rate model. Suppose there are M data rates, denoted $r_1, r_2, ..., r_M$, supported by the physical layer. Reliable communication by using rate r_m can be realized only if the signal strength at the receiver is above a certain threshold, say η_m. Consequently, for the set of M data rates, there is a set of M thresholds, $\eta_1, ..., \eta_M$. We further define $\eta_0 = 0$ and $\eta_{M+1} = \infty$.[2]

We study the case where the following large-scale propagation model is applicable: $P_2 = \frac{P_1}{d_2^\alpha}$, where P_1 is the reference signal power measured at one meter away from the transmitter, P_2 is the signal power measured at d_2 meters away from the transmitter, and α is a positive constant representing the path loss roll off factor. The reference power P_1 can be obtained via field measurement or calculated using the free space path loss formula in [12].

[2] In IEEE 802.11g, there are 11 different bit rates, and the minimal threshold for each bit rate is specified by the standard.

In addition to large scale propagation, multipath fading may have a prominent effect on reliable communication. Under Rayleigh fading, the instantaneous power, γ, is exponentially distributed with the probability density function $p(\gamma) = \frac{1}{P_r} e^{\frac{-\gamma}{P_r}}$, where P_r is the average power of γ. Consequently, the probability that a transmitter with reference power P can transmit at rate r_m, to a receiver at distance l, where $l > 1$, is $p(r_m, l, P) = \int_{\eta_m}^{\eta_{m+1}} \frac{l^\alpha}{P} e^{\frac{-\gamma l^\alpha}{P}} d\gamma$, where $m = 1, 2, ..., M$. Furthermore, in some instances, the receiver can be located in a deep-fade area, i.e., is experiencing bad channel condition. The probability of these instances, where the transmitter cannot transmit in any data rate, is $p_f(l, P) = \int_{\eta_0}^{\eta_1} \frac{l^\alpha}{P} e^{\frac{-\gamma l^\alpha}{P}} d\gamma$, while the probability that the transmitter can transmit successfully is $p_s(l, P) = 1 - p_f(l, P) = \sum_{m=1}^{M} p(r_m, l, P)$.

In this discrete-rate model with fading, the transmitter needs a small channel probing time, denoted T_{prob}, to test the channel and decide the transmission rate before the actual data transmission can take place. If the transmitter determines that the channel condition does not allow it to transmit at any rate, it will give up its transmit opportunity, and prob the channel again later. The wasted channel probing time adds to the total packet transmission time.

Let $T_g(l, P, x)$ be a random variable that represents the packet transmission time of an x-bit packet, and let S and F be the events of "good" and "bad" channel states respectively. The expected value of this packet transmission time is

$$\begin{aligned} & E[T_g(l, P, x)] \\ & = E[T_g(l, P, x)|F] p_f(l, P) + E[T_g(l, P, x)|S] p_s(l, P) \\ & = \Big(E[T_g(l, P, x)] + T_{prob} \Big) p_f(l, P) + \Big(\sum_{m=1}^{M} \frac{p(r_m, l, P)}{p_s(l, P)} \Big[\frac{x}{r_m} + T_{prob} \Big] \Big) p_s(l, P). \end{aligned} \tag{18}$$

Rearranging the above, we have the expected packet transmission time

$$T(l, P, x) = E[T_g(l, P, x)] = \frac{T_{prob}}{p_s(l, P)} + \sum_{m=1}^{M} \frac{p(r_m, l, P)}{p_s(l, P)} \frac{x}{r_m} \quad . \tag{19}$$

6 Numerical Analysis

In this section, we present numerical results from the proposed optimization methods and evaluate the capacity improvement from using wireless relay nodes in an urban WLAN. Unless otherwise stated, the system parameters such as bit rate power thresholds, antenna gains, and transmitter powers are taken from the CISCO Aironet 1100 Series Access Point and mobile network interface card specifications [13]. The other system parameters are selected based on a typical urban environment.

6.1 Convergence of Lagrangian Iteration

In Fig. 4, we show the convergence of the algorithm in two typical network scenarios. For both scenarios, the network provides a coverage area of 400 meters in radius, and 16

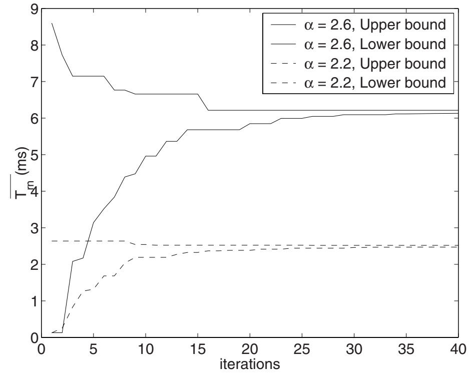


Fig. 4. Example convergence of the Lagrangian relaxation iterative algorithm

RNs are available to be placed in this network without restriction. MHs are uniformly distributed in the network coverage area. Both the AP and RN are equipped with a 10dBm transmitter, while the mobile hosts use a 5dBm transmitter. The combined length of a uplink and a downlink packet is set to 2k bytes, and 70% of downlink traffic is assumed. By default, the network is discretized into 100 thousand cells, corresponding to approximately 5 square meters per cell on average.

As shown in Fig. 4, for both channel roll-off factors, $\alpha = 2.2$ and $\alpha = 2.6$, the difference between the upper bound and the lower bound converges to less than 2% of the lower bound value in less than 40 iterations.

6.2 Effect of System Parameters on RN Placement and Performance Gain

In this subsection, we discuss the benefit of the strategically placed RNs with respect to different system parameters. The system that we investigated has three system parameters: roll off factor (α), proportion of downlink data (β), and the number of RNs (N). For each set of parameters, our analysis and optimization procedure produces an optimal placement of RNs. Moreover, as defined in (1), we can calculate the throughput capacity of the network without relay nodes, C_{norn} and with relay nodes optimally placed, C_{rn}. Hence, C_{norn} and C_{rn} are defined as $\frac{x}{\overline{T_{norn}}}$ and $\frac{x}{\overline{T_{rn}(\underline{d^*},\underline{\varphi^*})}}$ respectively, where $(\underline{d^*}, \underline{\varphi^*})$ represents the optimal RN location(s) calculated by the Lagrangian relaxation iterative algorithm. We define the performance gain of utilizing the RNs as $Gain = 100 \times \frac{C_{rn}-C_{norn}}{C_{norn}}$. In the following, the relationship among the three system parameters with respect to the optimal RN placement and performance gain will be discussed. Moreover, we also compare the optimal performance gain with the performance gains resulting from random placements of RNs.[3]

We study the effect of the roll off factor, α, and the number of RNs, N, in Fig. 5 and Fig. 6. Except N and α, the same set of system parameters from subsection 6.1 are used. Fig. 5 and Fig. 6 show the performance gains and optimal RN placements with respect

[3] For each set of system parameters, 100 different random RN placements were generated, and we report the average performance gain of relaying using randomly place RNs.

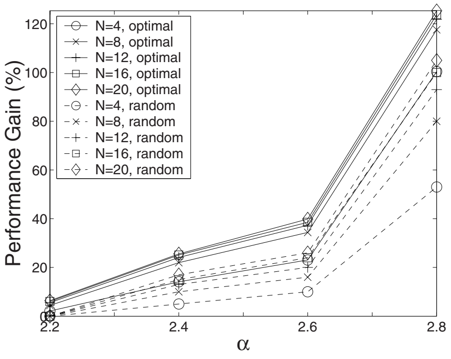


Fig. 5. Performance gain with respect to different roll-off factors and number of RNs

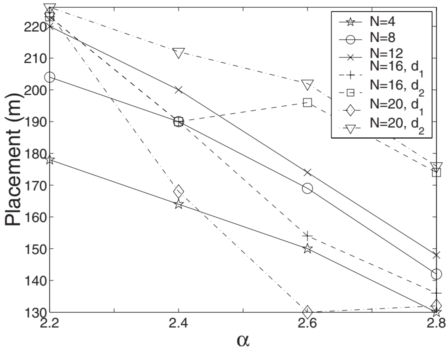


Fig. 6. Optimal placement of RNs with respect to different roll-off factors and number of RNs

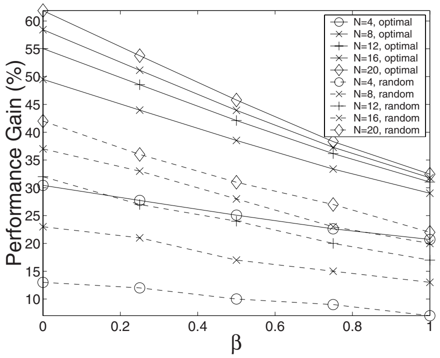


Fig. 7. Performance gain with respect to different proportion of downlink data and number of RNs

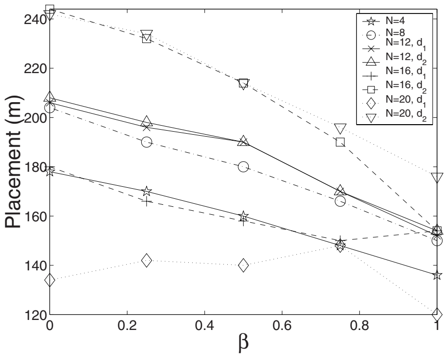


Fig. 8. Optimal placement of RNs with respect to different proportion of downlink data and number of RNs

to different roll off factors respectively. When there are 4, 8 or 12 RNs, the solution calculated by the Lagrangian algorithm converges to a single-tier configuration, where the RNs are uniformly distributed around and with a displacement d_1 meters away from the AP. When there are 16 or 20 RNs available, the Lagrangian algorithm converges to a two-tier configuration, where two equal-size rings of RNs with radius d_1 and d_2 are formed surrounding the AP.

From these figures, we make three main observations. First, the performance gain is high when the pathloss roll-off factor is high. The roll off factor determines how fast the signal decays when it travels through a distance. Therefore, as the roll off factor increases, the benefits of the RNs become more significant. Second, the performance gain difference between optimal and random placement of the RNs is substantial when the number of RN is small to moderate. Third, in all cases, the effect of diminishing return is observed as the number of RNs increases. These observations suggest that when the number of RN is high, the marginal gain of each addition RN is small.

We study the effect of the proportion of downlink data (β) and the number of RNs (N) on an urban network in Fig. 7 and Fig. 8. The system parameters are the same as before, except the roll off factor is set to 2.6. Again, the combined length of a uplink and a downlink packet is set to 2k bytes. Thus, the downlink and uplink packet lengths are 2βk bytes and $2(1-\beta)$k bytes respectively.

When there are 4 or 8 RNs, the Lagrangian algorithm converges to an single-tier configuration regardless of the proportion of downlink data. When there are 12, 16 or 20 RNs, the algorithm converges to a two-tier configuration, similar to the previous subsection.

Two main observations can be seen from these figures. First, the performance gain increases as the proportion of uplink data, $(1-\beta)$, increases. This is because the MH's transmitter has less power compared with that of the AP and RNs. As the amount of data needed to be transmitted by the MH's transmitter increases, the benefit of the RN becomes more significant. Second, relaying with optimally placed RNs performs significantly better than that of random placement regardless of traffic pattern.

7 Conclusions

In this work, we have investigated the strategic placement of wireless relay nodes to enhance the throughput capacity of an urban wireless local area network. We have developed an analytical model for performance evaluation and RN placement optimization. We propose a Lagrangian relaxation iterative algorithm to solve a discrete version of the RN placement problem. The proposed framework can be generalized to fit different channel models, network configurations, and user behaviors. In particular, we have investigated the RN placement problem in an IEEE 802.11g multi-rate WLAN under Rayleigh fading. Using the proposed numerical analysis framework, we have showed that in most cases, by using strategically placed RNs, the network capacity can be significantly improved. Given a set of network parameters, the proposed algorithm can be used by network designers to compute the optimal placement of RNs and justify the tradeoff between additional hardware cost and system performance gain.

References

1. Haas, Z.J., Deng, J., Liang, B., Papadimitratos, P., Sajama, S.: Wireless ad hoc networks. In Proakis, J., ed.: Wiley Encyclopedia of Telecommunications. John Wiley & Sons (2002)
2. Lin, Y., Hsu, Y.: Multihop cellular: A new architecture for wireless communications. In: Proc. of IEEE INFOCOM. (2000) 1273 – 1282
3. Cruz, R., Santhanam, A.: Optimal routing, link scheduling and power control in multihop wireless networks. In: Proc. of IEEE INFOCOM. (2003) 702–711
4. Mengesha, S., Karl, H., A.Wolisz: Capacity increase of multi-hop cellular WLANs exploiting data rate adapatation and frequency recycling. Technical report, Technical University Berlin Telecommunication Networks Group (2003)
5. Zhu, H., Cao, G.: rDCF: A relay-enabled medium access control protocol for wireless ad hoc networks. In: Proc. of IEEE INFOCOM. (2005) 12–22
6. Zhu, H., Cao, G.: On improving the performance of IEEE 802.11 with relay-enabled PCF. ACM/Kluwer Mobile Networking and Applications (MONET) **9** (2004) 423–434

7. Wu, H., Qiao, C., De, S., Tonguz, O.: Integrated cellular and ad hoc relaying systems: iCAR. IEEE Journal on Selected Areas in Communications **19**(10) (2001) 2105–2215
8. So, A., Liang, B.: Effect of relaying on capacity improvement in wireless local area networks. In: Proc. of IEEE WCNC. (2005) 1539 – 1544
9. So, A., Liang, B.: An efficient algorithm for the optimal placement of wireless extension points in rectilineal wireless local area networks. In: Proc. of International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine). (2005) 25–33
10. Daskin, M.: Network and Discrete Location: Models, Algorithms and Applications. John Wiley & Sons (1995)
11. Martin, R.K.: Large Scale Linear and Integer Programming. Kluwer Academic Publishers (1999)
12. Rappaport, T.S.: Wireless Communications: Principles and Practice. Prentice Hall (2001)
13. CISCO: Aironet 1100 series access point: Data sheet. Technical report, CISCO Systems (2003)

Correlated Equilibrium in Access Control for Wireless Communications

Eitan Altman[1], Nicolas Bonneau[1], and Mérouane Debbah[2]

[1] INRIA, Centre Sophia Antipolis, 2004 Route des Lucioles, B.P.93,
06902 Sophia Antipolis, France
{eitan.altman, nicolas.bonneau}@sophia.inria.fr
[2] Mobile Communications Group, Institut Eurecom, 2229 Route des Cretes, B.P.193,
06904 Sophia Antipolis, France
merouane.debbah@eurecom.fr

Abstract. We study a finite population of mobiles communicating using the slotted ALOHA-type protocol. Our objective is the study of coordination between the mobiles in both cooperative as well as non-cooperative scenarios. Our study is based on the correlated equilibrium concept, a notion introduced by Aumann that broadens the Nash equilibrium. We study ways in which signaling can improve the performance both in the cooperative as well as in the non-cooperative cases, even in the absence of any extra information being conveyed through these signals.

1 Introduction

There has been a growing interest in studying competition (and also cooperation) aspects of networking in general, and of access to a common channel in particular. Non-cooperative game theory, as well as cooperative game theory, have been frequently used as a central framework for modeling such issues, see for example [1] and references therein.

We consider a finite population of mobile terminals that compete over the access to a common channel. The framework we consider is of a discrete time system (a simplified version of the slotted Aloha protocol [2]). All mobiles are thus supposed to be synchronized. As is frequently assumed when studying slotted Aloha, we assume that if more than one mobile attempts to send a packet at time slot t then all transmitted packets are lost and mobiles wait a random amount of slots before retransmitting their packets, in order to avoid repeated collisions.

We consider both the cooperative as well as the non-cooperative approaches. For each case we study the impact of adding coordination mechanisms on the throughput.

The framework we consider is that of correlated games along with the notion of *correlated equilibrium.* The notion of correlated equilibrium was introduced by R. Aumann[1] in [3] and further studied in [4, 5, 6]. An algorithm for the

[1] Prof. R. Aumann has received in 2005 the Nobel prize in economy for his contributions to game theory, together with Thomas Schelling.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 173–183, 2006.

computation of correlated equilibria is developed in [7]. Correlated equilibria are generalizations of the Nash equilibrium concept; the correlated equilibria are defined in a context where there is an arbitrator who can send (private or public) signals to the players. These signals allow players to coordinate their actions, and, in particular, to perform joint randomization over strategies.

In many contexts, an arbitrator is thought of as an intelligent entity, used for helping to solve conflicts and for proposing compromises to the different sides involved. In contrast, in correlated games, an arbitrator needs not have any intelligence. It is assumed to generate signals that do not depend on the system (or on individual) states. Moreover, it does not need to have any knowledge on the system. All the arbitrator has to do is to create some random signals (according to a randomized mechanism known by the players) that can help the synchronization (or coordination) between them.

In the context of non-cooperative games, each player has the possibility not to consider the signal(s) it receives. A multi-strategy obtained using the signals is a set of strategies (one strategy for each player which may depend on all the information available to the player including the signal it receives). It is said to be a correlated equilibrium (a precise definition will be given later) if no player has an incentive to deviate unilaterally from its part of the multi-strategy. A special type of "deviation" in this definition can be of course to ignore the signals.

An arbitrator may even be a virtual entity. As an example, the players can agree to use some random data (e.g., the first word they hear on the radio) as the signal or as an input to a function that allows to create a common signal (or a signal which may differ from one player to another).

Our contribution in this paper is not only in applying the notion of correlated equilibrium in the context of networking, but also in extending it to the multi-criterion case; in our case, each mobile (player) has two objectives: expected throughput and expected power consumption. We use the correlated equilibrium setting adapted to the context of constrained optimization by each player (maximizing the average throughput with a constraint on the average power consumption).

Coordination between players turns out to be useful also in the case of cooperative optimization. Indeed, the coordination may be needed also in this framework in the so called *team problem* [1, 8], i.e., where various players have the same common objective that they maximize (e.g., the global throughput). Users may benefit from performing joint randomizations, which may not be possible without coordination due to a possible distributed nature of the problem. The need for joint randomization in the team setting is due to the multi-objective nature of the problem (more precisely to the constraints on the expected power consumption).

The paper is organized as follows. The model is described in Sect. 2. The general game is analyzed in Sect. 3. We introduce a coordination mechanism, and we define and analyze the corresponding correlated strategies that arise in this context in Sect. 4. Finally, we present some results in Sect. 5.

2 The Model

We consider a finite population of m mobile terminals. Each mobile has a unique i.d. number ranging between 1 to m. Time is slotted.

Let $\mathcal{N} = \{0,1\}^m$ represent the set of all 2^m subsets of $\{1,\ldots,m\}$. At each time slot, a subset of mobiles $\mathbf{Z}(t) \in \mathcal{N}$ is assumed to be active. The number of active terminals at time t is equal to the Hamming weight $|\mathbf{Z}(t)| = \sum_{i=1}^m Z_i(t)$ of $\mathbf{Z}(t)$ and denoted by $N(t)$. $\mathbf{Z}(t)$ (and thus $N(t) = |\mathbf{Z}(t)|$) are assumed to be stationary ergodic processes.

Each active mobile is assumed to be saturated, i.e., it always has packets to send. At each time slot, a random subset of mobiles is active. If at a time slot, more than one active mobile attempts to transmit then there is a collision and all packets transmitted in the time slot are lost.

Let q_i denote the probability that mobile i transmits a packet when active (we call q_i the *strategy*). If $\mathbf{z} \in \mathcal{N}$, let $\zeta(\mathbf{z})$ be the probability that the subset $\mathbf{z}$ of mobiles is active at a slot and let $\pi_n = \sum_{|\mathbf{z}|=n} \zeta(\mathbf{z})$ be the probability that there are n active mobiles at a slot. In particular, the probability that mobile i is the only active mobile in a slot is $\zeta(\mathbf{e}_i)$ where $\mathbf{e}_i$ is the vector whose elements are all zero except for the ith entry which equals one.

The probability of a successful transmission at a time slot is

$$\begin{aligned}\Theta_{\text{all}}(q_1,\ldots,q_m) &= \mathbb{E}_{\mathbf{Z}}\left[\sum_{i\in\mathbf{Z}} q_i \prod_{j\in\mathbf{Z}\setminus\{i\}} (1-q_j)\right] \\ &= \sum_{\mathbf{z}\in\mathcal{N}} \zeta(\mathbf{z}) \sum_{i\in\mathbf{z}} q_i \prod_{j\in\mathbf{z}\setminus\{i\}} (1-q_j). \qquad (1)\end{aligned}$$

which is also the system throughput. The expected average throughput per mobile is Θ_{all}/m. The throughput of mobile i conditioned on being active is given by

$$\begin{aligned}\Theta_i^{\text{act}}(q_1,\ldots,q_m) &= \mathbb{E}_{\mathbf{Z}}\left[q_i \prod_{j\in\mathbf{Z}\setminus\{i\}} (1-q_j) \,\middle|\, i\in\mathbf{Z}\right] \\ &= q_i \sum_{\substack{\mathbf{z}\in\mathcal{N}\\ i\in\mathbf{z}}} \zeta(\mathbf{z}) \prod_{j\in\mathbf{z}\setminus\{i\}} (1-q_j). \qquad (2)\end{aligned}$$

In the following, the purpose of cooperative optimization will be to maximize the system throughput Θ_{all}, whereas in a non-cooperative setting, each mobile will attempt to maximize selfishly its conditional throughput Θ_i^{act}, which we call its *utility*.

3 No Coordination Mechanism

3.1 General Case

The maximal throughput that can be attained is obtained by maximizing the system throughtput $\Theta_{\text{all}}(q_1,\ldots,q_m)$ given by (1) over $(q_1,\ldots,q_m) \in [0,1]^m$. Since

$\Theta_{\mathrm{all}}(q_1, \ldots, q_m)$ is a multivariate polynomial, hence continuous in $(q_1, \ldots, q_m)$, and $[0,1]^m$ is a compact set, the existence of a maximum is immediate. For given $\{\zeta(\mathbf{z})/\mathbf{z} \in \mathcal{N}\}$, computing this maximum is a constrained optimization problem [9].

If the mobiles are non-cooperative and care only for their own throughput then it is immediate from (2) that the only Nash equilibrium[2] is where all mobiles transmit with $q_i = 1$. The global throughput is then π_1 and the expected average throughput per mobile is π_1/m.

In the non-cooperative case, we are also interested by the *conditional* throughput, i.e., the throughput of a mobile averaged over the activity periods of the mobile. The conditional throughput of mobile i when $q_i = 1$ for all mobiles is given by $\zeta(\mathbf{e}_i)$.

3.2 Power Considerations

In reality mobile users are sensitive to power consumption. Their objective is to maximize the system throughput (in the cooperative case) or the individual throughput (in the non-cooperative case) under the constraints $q_i \le q_i^{\max}$ for some constant $q_i^{\max}$, for all users i. In the cooperative case, we can model the choice of transmission probability q_i as a constrained optimization problem. In the non-cooperative case, it is easy to see that the Nash equilibrium is obtained with $q_i = q_i^{\max}$ for all mobiles. From (1), this gives at the Nash equilibrium the throughput of

$$\Theta_{\mathrm{all}}(q_1^{\max}, \ldots, q_m^{\max}) = \sum_{\mathbf{z}\in\mathcal{N}} \zeta(\mathbf{z}) \sum_{i\in\mathbf{z}} q_i^{\max} \prod_{j\in\mathbf{z}\setminus\{i\}} (1 - q_j^{\max}),$$

and from (2), the conditional throughput as

$$\Theta_i^{\mathrm{act}}(q_1^{\max}, \ldots, q_m^{\max}) = q_i^{\max} \sum_{\substack{\mathbf{z}\in\mathcal{N}\\ i\in\mathbf{z}}} \zeta(\mathbf{z}) \prod_{j\in\mathbf{z}\setminus\{i\}} (1 - q_j^{\max}).$$

4 Coordination, Correlated Equilibrium and Optimization

4.1 Coordination Mechanism

If the base station had full information and could schedule transmissions of the mobiles then full utilisation (i.e., a throughput of $1 - \pi_0$) could be achieved by a TDMA type approach. We consider however the case where the base station has no control over the mobiles and has no information on their power constraints nor on their number. It can only serve as an arbitrator, in the sense that was discussed in the introduction.

[2] A Nash equilibrium is a set of strategies such that no mobile can improve its utility by deviating unilaterally from its strategy.

We therefore consider the following coordination mechanism. We assume that at each time slot t, the base station can send a signal to all mobiles in the form of a random variable $X(t)$, uniformly distributed over the integers $\{0, \ldots, K-1\}$ for some integer $K \geq 2$. We assume for simplicity that m is a multiple of K. The process $X(t)$ is assumed to be independent of $\mathbf{Z}(t)$.

4.2 Transmission Strategy for Mobiles

In absence of any coordination mechanism, a strategy of a mobile would be the probability of transmitting a packet. In the presence of the coordination mechanism, a mobile has the possibility to use a larger notion of strategies.

Definition 1. *We define the set of correlated policies as follows.*

- *We partition the set of all mobiles into K subgroups S_j, $j = 1, \ldots, K$ where S_j contains a mobile i if and only if $i = j - 1 \pmod{K}$ (denoted $i \equiv j - 1$).*
- *A correlated strategy of a mobile is described using two real numbers in the unit interval: p_i and q_i.*
- *At time t, an active mobile i transmits a packet with probability p_i if and only if $i \in S_{X(t)}$. Otherwise it transmits with probability q_i.*

Note that this class of correlated strategies includes in particular the non-correlated strategies. Thus, in the non-cooperative setting, a mobile has always the possibility of ignoring the signals $X(t)$ by using $p_i = q_i$. The latter can be viewed as a non-correlated strategy.

We call (p_i, q_i) the strategy of mobile i. For two m-dimensional vectors $\mathbf{p}$ and $\mathbf{q}$ we define $(\mathbf{p}, \mathbf{q})$ to be a multi-strategy for all mobiles, where mobile i uses the ith entry (p_i, q_i) of the vectors $(\mathbf{p}, \mathbf{q})$. Let

$$\mathcal{U} = \{(\mathbf{p}, \mathbf{q}) / \forall i \in \{1, \ldots, m\},\ p_i \in [0,1],\ q_i \in [0,1]\}$$

denote the class of all multi-strategies.

Define $(\mathbf{p}, \mathbf{q})^{-i}$ to be the set of $m-1$ strategies of all mobiles except for mobile i, and set $\Big((\mathbf{p}, \mathbf{q})^{-i}, (p', q')_i\Big)$ to be the policy where all mobiles other than the ith one use the policies described by $(\mathbf{p}, \mathbf{q})^{-i}$ whereas the ith mobile uses policy (p', q').

4.3 Power Considerations

We assume that mobile i has a constraint on the average power it can use while active. More precisely, the average power consumption during activity periods of a mobile with parameters (p, q) is

$$\mathrm{Pow}(p, q) = \frac{p}{K} + \frac{(K-1)q}{K}. \tag{3}$$

We then assume that mobile i has the power constraint

$$\mathrm{Pow}(p_i, q_i) \leq q_i^{\max} \text{ where } q_i^{\max} \leq 1. \tag{4}$$

Let U_i^{cons} denote the class of strategies of mobile i satisfying (4). Let

$$\mathcal{U}^{\text{cons}} = \{\mathbf{u} \in \mathcal{U} / \forall i \in \{1, \ldots, m\},\ u_i \in U_i^{\text{cons}}\}$$

denote the class of multi-strategies $\mathbf{u}$ for which for each i, $u_i = (p_i, q_i)$ satisfies (4).

Definition 2. *A multi-strategy* $\mathbf{u} \in \mathcal{U}^{cons}$ *is said to be a correlated equilibrium if for all* i *and* $(p', q') \in U_i^{cons}$

$$\Theta_i^{act}(\mathbf{u}) \geq \Theta_i^{act}\Big(\mathbf{u}^{-i}, (p', q')_i\Big). \tag{5}$$

Definition 3. *A multi-strategy* $\mathbf{u}^* \in \mathcal{U}^{cons}$ *is said to be correlated optimal if for all feasible multi-strategies* $\mathbf{u} \in \mathcal{U}^{cons}$*,*

$$\Theta_{all}(\mathbf{u}^*) \geq \Theta_{all}(\mathbf{u}). \tag{6}$$

The expressions for $\Theta_{\text{all}}(\mathbf{u})$ and $\Theta_i^{\text{act}}(\mathbf{u})$ can be written as

$$\begin{aligned}\Theta_{\text{all}}(\mathbf{u}) = \sum_{\mathbf{z} \in \mathcal{N}} \zeta(\mathbf{z}) \sum_{i \in \mathbf{z}} \Bigg(& \frac{p_i}{K} \prod_{\substack{j \in \mathbf{z} \setminus \{i\} \\ j \equiv i}} (1 - p_j) \prod_{\substack{j \in \mathbf{z} \setminus \{i\} \\ j \not\equiv i}} (1 - q_j) \\ & + \frac{q_i}{K} \sum_{\substack{k=1 \\ k \not\equiv i}}^{K} \prod_{\substack{j \in \mathbf{z} \setminus \{i\} \\ j \equiv k}} (1 - p_j) \prod_{\substack{j \in \mathbf{z} \setminus \{i\} \\ j \not\equiv k}} (1 - q_j) \Bigg)\end{aligned} \tag{7}$$

and

$$\begin{aligned}\Theta_i^{\text{act}}(\mathbf{u}) = & \frac{p_i}{K} \sum_{\substack{\mathbf{z} \in \mathcal{N} \\ i \in \mathbf{z}}} \zeta(\mathbf{z}) \prod_{\substack{j \in \mathbf{z} \setminus \{i\} \\ j \equiv i}} (1 - p_j) \prod_{\substack{j \in \mathbf{z} \setminus \{i\} \\ j \not\equiv i}} (1 - q_j) \\ & + \frac{q_i}{K} \sum_{\substack{\mathbf{z} \in \mathcal{N} \\ i \in \mathbf{z}}} \zeta(\mathbf{z}) \sum_{\substack{k=1 \\ k \not\equiv i}}^{K} \prod_{\substack{j \in \mathbf{z} \setminus \{i\} \\ j \equiv k}} (1 - p_j) \prod_{\substack{j \in \mathbf{z} \setminus \{i\} \\ j \not\equiv k}} (1 - q_j)\,.\end{aligned} \tag{8}$$

$\Theta_i^{\text{act}}(\mathbf{u})$ is an affine function of p_i and q_i. Therefore, in order to maximize $\Theta_i^{\text{act}}(\mathbf{u})$, the inequality in (4) will be an equality: each mobile will transmit at the maximum of its possibilities. In the next section, we investigate how the power is split between p_i and q_i for each mobile in a particular case.

4.4 Symmetric Case

Solving the constrained optimization problems of (1) or (7), as well as finding Nash or correlated equilibria, becomes rapidly intractable in the general case when the number of mobiles m (and hence the number of variables in the multivariate polynomials involved) increases. To simplify the analysis, we consider a symmetric case when the coefficients $\zeta(\mathbf{z})$ depend only on $|\mathbf{z}|$, and the power constraints $q_i^{\max} = q^{\max}$ are the same for all users.

We consider a simple model when mobiles are independently active with a probability π. This corresponds to the model used in [10] for users with a single packet buffer, when the probability of arrival of a new packet is equal to the probability of retransmission of a backlogged packet. In this case, the coefficients in (1) and (7) become symmetric, since for all $\mathbf{z}$ such that $|\mathbf{z}| = n$, $\zeta(\mathbf{z})$ are equal:

$$\zeta(\mathbf{z}) = \pi^{|\mathbf{z}|}(1-\pi)^{m-|\mathbf{z}|} \tag{9}$$

In the non-cooperative case, we can restrict to the same strategy (p, q) being used by all users, and investigate if a single user deviating from this strategy benefits by using a different strategy $(\hat{p}, \hat{q})$. Recall that $\pi_n = \sum_{|\mathbf{z}|=n} \zeta(\mathbf{z})$. Let

$$\ell = \frac{m}{K}, \qquad \lambda = m - \frac{m}{K}.$$

After some manipulations, (8) can be rewritten as:

$$\begin{aligned}
\Theta^{\mathrm{act}}(\mathbf{u}) = {} & \frac{\hat{p}}{K} \sum_{n=1}^{m} \frac{1}{\binom{m}{n}} \pi_n \\
& \times \sum_{k=\max(0,n-1-\lambda)}^{\min(\ell-1,n-1)} \binom{\ell-1}{k} \binom{\lambda}{n-1-k} (1-p)^k (1-q)^{n-1-k} \\
& + \frac{(K-1)\hat{q}}{K} \sum_{n=1}^{m} \frac{1}{\binom{m}{n}} \pi_n \\
& \times \sum_{k=\max(0,n-\lambda)}^{\min(\ell,n-1)} \binom{\ell}{k} \binom{\lambda-1}{n-1-k} (1-p)^k (1-q)^{n-1-k}.
\end{aligned} \tag{10}$$

The power constraints (4) give us

$$\hat{p} = Kq^{\max} - (K-1)\hat{q}. \tag{11}$$

Replacing $\hat{p}$ by this expression in (10), we obtain $\Theta^{\mathrm{act}}(\mathbf{u})$ as an affine function in $\hat{q}$. Hence, the optimal $\hat{q}$ will be either

$$\max(0, \frac{Kq^{\max}-1}{K-1}) \text{ or } \min(1, \frac{Kq^{\max}}{K-1})$$

depending on the sign of the coefficient

$$\begin{aligned}
& \sum_{n=1}^{m} \frac{1}{\binom{m}{n}} \pi_n \sum_{k=\max(0,n-\lambda)}^{\min(\ell,n-1)} \binom{\ell}{k} \binom{\lambda-1}{n-1-k} (1-p)^k (1-q)^{n-1-k} \\
& - \sum_{n=1}^{m} \frac{1}{\binom{m}{n}} \pi_n \sum_{k=\max(0,n-1-\lambda)}^{\min(\ell-1,n-1)} \binom{\ell-1}{k} \binom{\lambda}{n-1-k} (1-p)^k (1-q)^{n-1-k}.
\end{aligned} \tag{12}$$

This gives us a simple formula to investigate whether or not a given value of (p, q) that saturates (4) is a correlated equilibrium: replace (p, q) by their values in (12) and estimate the sign of the expression. If the chosen q satisfies

$$q = \max(0, \frac{Kq^{\max} - 1}{K - 1})$$

and the sign of (12) is negative or if the chosen q satisfies

$$q = \min(1, \frac{Kq^{\max}}{K - 1})$$

and the sign of (12) is positive, then (p, q) is indeed a correlated equilibrium.

5 Results

We use the terms *cooperative* and *non-cooperative* to describe the behavior of mobiles, whereas the term *coordination* refers to the presence of a common signal. Without coordination, the equilibrium concept in the non-cooperative case is the *Nash equilibrium*, whereas it is the *correlated equilibrium* with coordination.

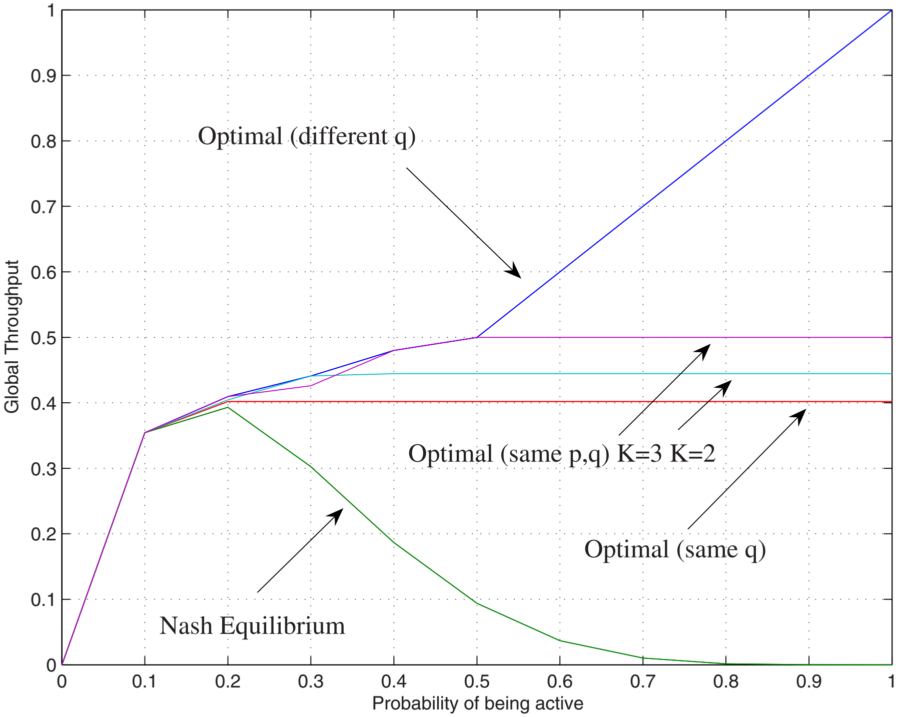


Fig. 1. System throughput versus the probability of being active for a mobile with and without coordination, for 6 users, without power constraints

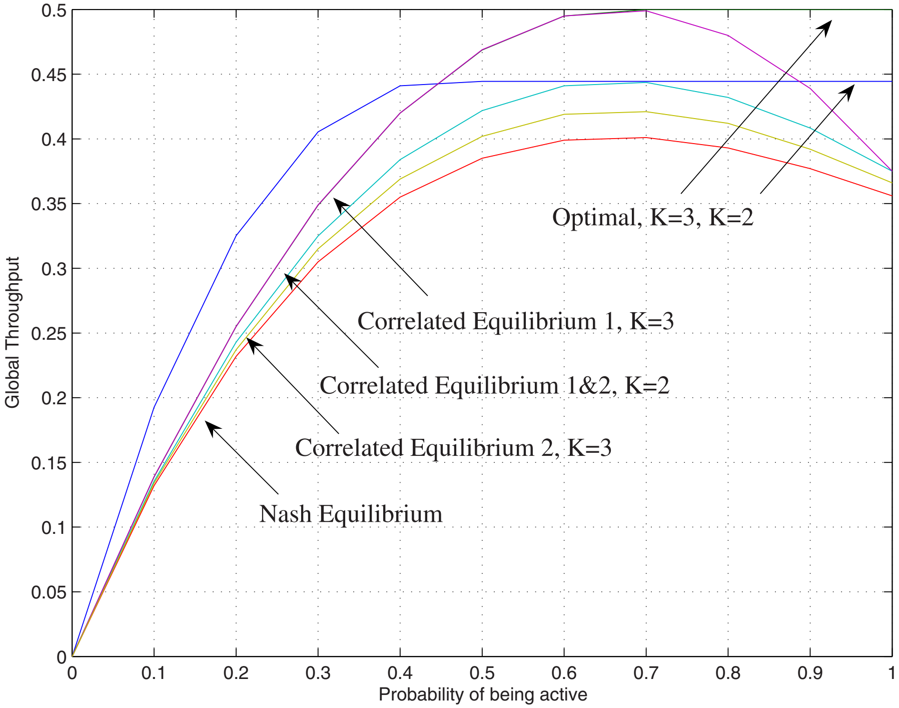


Fig. 2. System throughput versus the probability of being active for a mobile with power constraint $q^{\max} = 0.25$, for 6 users

In this section, we consider the setting of Subsect. 4.4. However, an interesting result is that, even in this symmetric case, the optimal throughput is neither reached by saturating the power constraints $q_i^{\max}$ for all users nor for a symmetric attribution of the channel (i.e., the same strategy for all users).

In Fig. 1, we have plotted the system throughput Θ_{all} versus the probability of being active π with and without coordination, according to (1) and (7), for 6 users, without power constraints ($q_i^{\max} = 1$ for all users). We observe that the optimal throughput with the same strategy for all mobiles reaches a plateau and stays constant, no matter how active the mobiles are. With coordination, the value of this plateau is increased.

With a non-symmetric attribution of the strategies, a higher system throughput can be achieved. The linear portion of the curve, for $\pi > 0.5$, is actually obtained by letting only one user transmit; for $\pi \leq 0.5$, it is optimal to let several users transmit. Without power constraints, the optimal throughput with coordination is the same as without coordination.

The system throughput reached at Nash equilibrium (i.e., $q = 1$ for all mobiles) is close to the optimum for low values of π (when few mobiles are active), but rapidly decreases and approaches 0 as the probability of being active increases. Note that without power constraints, Nash and correlated equilibrium coincide, therefore the coordination mechanism does not increase the throughput in the non-cooperative case. We remark that the curve for Nash equilibrium corresponds to the throughput calculated in [10], which is simply $m\pi(1-\pi)^{m-1}$.

The optimal curves in Fig. 1 are obtained without power constraints, therefore with power constraints the optimal curves will always be lower.

In Fig. 2, we have plotted the system throughput Θ_{all} obtained in the correlated equilibrium with power constraint $q^{\max} = 0.25$. In the case $K = 3$, two correlated equilibria are possible: $p = 0.75$, $q = 0$ or $p = 0$, $q = 0.375$ (denoted respectively as 1 and 2 in the figure). In the case $K = 2$, there are two correlated equilibria as well: $p = 0.5$, $q = 0$ and $p = 0$, $q = 0.5$ (both give the same system throughput). As a comparison, we have plotted the optimal throughput that can be obtained under the power constraint $q^{\max} = 0.25$ in the cooperative case, as well as the throughput obtained in the Nash equilibrium without coordination.

Non-cooperative throughput is improved compared to the case without power constraints. With strong power constraints, we observe that the coordination mechanism allows to obtain higher values of the throughput in the non-cooperative case. For some probabilities π, non-cooperative global throughput almost reaches the values obtained in the cooperative case.

6 Conclusion

We have investigated a game theoretical setting including a coordination mechanism in a distributed access control. Our analysis is based on the concept of correlated equilibrium that enriches the strategies of mobiles. Power constraints are primordial in order to give a sense to coordination. In the absence of power constraints, coordination does not necessarily improve the channel utilization. The proposed coordination mechanism can improve the utilization of the channel in presence of power constraints, even in presence of selfish users.

References

1. Altman, E., Boulogne, T., Azouzi, R.E., Jimenez, T., Wynter, L.: A survey on networking games. Computers and Operations Research (2005)
2. Roberts, L.: Aloha Packet System with and without Slots and Capture. Technical report, Stanford Research Institute, Advanced Research Projects Agency, Network Information Center (1972)
3. Aumann, R.: Subjectivity and Correlation in Randomized Strategies. Journal of Mathematical Economics **1** (1974) 67–96
4. Aumann, R.: Correlated Equilibrium as an Expression of Bayesian Rationality. Journal of Mathematical Economics **55** (1987) 1–18
5. Hart, S., Schmeidler, D.: Existence of Correlated Equilibria. Mathematics of Operations Research **14**(1) (1989) 18–25
6. Neyman, A.: Correlated Equilibrium and Potential Games. International Journal of Game Theory **26** (1997) 223–227
7. Papadimitriou, C.H.: Computing Correlated Equilibria in Multiplayer Games. (Available on the author's home page: `http://www.cs.berkeley.edu/~christos`)

8. Başar, T., Cruz, J.B. In: Concepts and methods in multiperson coordination and control. North-Holland Publishing Company (1982) 351–394
9. Luenberger, D.: Linear and Nonlinear Programming. 2nd edn. Addison-Wesley, Inc., Reading, Massachusetts (1984)
10. Kleinrock, L., Lam, S.: Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation. IEEE Trans. on Communications (4) (1975) 410–423

Design and Analysis of an Adaptive Backoff Algorithm for IEEE 802.11 DCF Mechanism

Mouhamad Ibrahim and Sara Alouf

INRIA – B.P. 93 – 06902 Sophia Antipolis – France
{mibrahim, salouf}@sophia.inria.fr

Abstract. This paper presents an adaptive backoff algorithm for the contention-based Distributed Coordination Function (DCF) of the IEEE 802.11 standard. Relying on on-line measurements of the number of sources, the algorithm, called Adaptive BEB, judiciously sets the size of the minimal contention window to adapt to the congestion level in the shared medium. The paper also provides an extension to Adaptive BEB for enhancing its performance over noisy channels. In this extension, a simple EWMA filter is used to derive a Packet Error Rate estimator. The performance evaluation of our proposal is addressed via simulations.

Keywords: DCF mechanism, optimal adaptation, performance evaluation, noise.

1 Introduction

Since its initial appearance in 1997, the IEEE 802.11 standard have engendered several research activities due to wireless local area networks (WLANs) specific features. One of the well-known drawbacks of IEEE 802.11 is the low performance of its MAC protocol in terms of throughput in congested networks. Several papers have pointed out this problem and proposed solutions to counter it. However, and as will be shown later, the performance of some of these degrades substantially under various scenarios.

In this paper, we aim at designing a new adaptive and robust backoff algorithm that enhances the performance of the Distributed Coordination Function (DCF) of the IEEE 802.11 standard under a wide variety of conditions. This new algorithm is meant to be simple and as close as possible to the standard. Our objectives are: maximum system performance in terms of maximum system throughput and minimum packet delay, and robustness to channel errors. Considering saturation conditions, i.e. active nodes have always packets to transmit, we derive a simple expression relating the minimum contention window size of the DCF backoff to the optimal transmission probability. The latter probability has been computed in [1], and an alternative formulation can be derived from [2, 3]. Both formulations require an estimate of the number of contending stations. In this paper, we propose a novel method for estimating the number of sources, which relies on counting *signs of life* coming from other stations to estimate their number. Despite many studies concerned with the optimization of the DCF in congested networks, the presence of errors on the channel is often disregarded. Most proposals rely on missed acknowledgments (ACK) from the destination to trigger the control of the contention window. However, missed ACKs are due either to congestion or channel

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 184–196, 2006.

errors. In this paper, we introduce a mechanism based on an Exponentially Weighted Moving Average (EWMA) estimator of the Packet Error Rate (PER) seen on the channel to adjust the contention window, reducing thus the overhead introduced by the noise while still avoiding collisions.

The rest of the paper is as follows: Sect. 2 briefly reviews the DCF and some related work. The analytical background needed to design our algorithm is presented in Sect. 3. Section 4 is devoted to the algorithm itself, motivating every step of it and describing its design and implementation. A simple estimator of the number of contending nodes is presented in Sect. 5 and an enhancement of the backoff algorithm in noisy environments is presented in Sect. 6. Section 7 presents the simulation results and Sect. 8 studies the fairness of our algorithm. Finally, Sect. 9 summarizes our work.

2 Preliminaries on the DCF Mechanism and Related Work

The Distributed Coordination Function (DCF) is the primary access protocol in 802.11 for sharing the wireless medium between active stations. In DCF, the stations listen to the channel before transmitting: upon channel idleness for a duration greater than the Distributed InterFrame Space (DIFS) period, the station transmits directly; otherwise it waits for channel idleness. When the channel becomes idle again for a DIFS, the station enters a deferring phase by selecting a random number of time slots in the range $[0, CW - 1]$ that is used to initialize a *backoff counter* [4]. Here CW refers to Contention Window and it is an integer, set at the first packet transmission attempt to the minimum contention window CW_0, and doubled as long as the transmission fails until reaching a maximum size $CW_{\max} = 2^m CW_0$, where m denotes the maximum backoff stage. The process of doubling the size of CW is called binary exponential backoff. The backoff counter is a timer decreased as long as the channel is sensed idle. When the backoff counter reaches 0, the station transmits directly, then waits for an ACK from the destination station. If the source station has not received an ACK within a specified ACK timeout, it will assume that the transmitted packet was lost due to a collision, and it will start the backoff procedure again after *doubling* its contention window size. On the other hand, if the packet is well received by the destination station, the latter will send an ACK to the source station that upon its reception will reset its contention window size to the minimal value CW_0 and proceed to the next packet in the buffer. In addition to this two-way handshaking technique, named basic access mechanism, DCF specifies another optional four-way handshaking technique called Request To Send/Clear To Send (RTS/CTS). The idea is to reserve the channel prior to a transmission by exchanging the RTS and CTS control frames between the source and the destination before the transmission of the packet.

The binary exponential backoff algorithm used in DCF has two major drawbacks. First, the contention window is increased upon transmission failure regardless of the cause of failure. Second, after a successful transmission of a packet, the contention window is reset to CW_0, thus forgetting its knowledge of the current congestion level in the network. These two points are at the basis of the inefficiency of the DCF mechanism.

As mentioned earlier, several research works have proposed modifications to IEEE 802.11 back-off algorithm to enhance its operation in congested networks. The earliest proposal [5] is based on a unique contention window CW, whose size is updated after each transmission attempt given an estimation of the number of active stations N. In order to estimate N, the authors measure the number of busy slots observed during a backoff decrease period. Another algorithm based on a unique contention window size is given in [2]. The authors establish an analogy between the standard protocol and the p-persistent IEEE 802.11 protocol, in which the backoff interval is sampled geometrically with parameter p. By maximizing the analytical expression found for the throughput of the p-persistent protocol, the authors derive the optimal p and subsequently the optimal size of the backoff interval. An estimation of the number of actual active users is required in this approach and the authors rely on measures of idle time to perform this estimation. In [6], the authors propose an extension to the DCF mechanism in order to achieve maximum throughput. Instead of directly transmitting when the backoff counter reaches 0, the authors propose to differ the transmission with a certain probability that depends, among other terms, on the slot utilization and the distribution of the packet lengths (assumed to be geometric). For this algorithm to work, each station requires a measure of the slot utilization and an inference of the packet length distribution. The authors of [7] propose to multiplicatively decrease the contention window after a successful transmission using a decrease factor δ in the range $[0, 1]$ that is set constant for all stations. The choice of δ greatly influences the performance of the algorithm. Maximum system throughput is achieved when $\delta = 0.9$ at the cost of a high system response time. A linear increase/linear decrease contention window algorithm is proposed in [8]. More precisely, upon a missed ACK, the current contention window is increased by a constant value ω. Upon receiving an ACK, it is decreased by ω with probability $1 - \delta$, and kept unchanged with probability δ. The parameters ω and δ are set heuristically to constant values. To the best of our knowledge, only [9] proposes a mechanism to enhance the DCF mechanism for noisy environments. The proposed algorithm resets the contention window when a failed transmission is assumed to be noise-corrupted. When the RTS/CTS mechanism is used, it considers as noise-corruption the absence of ACK and as collision the absence of CTS. When the basic access mechanism is used, the assumed noise-corruption probability is adjusted linearly to approach the ideal transmission probability, computed thanks to a count of the number of active stations. Last, we would like to briefly mention that [10] proposes a Kalman filter to estimate the number of active users N and relies on channel sensing to measure the collision probability (noisy channels are not considered in this study).

3 The Model

To design our algorithm, we adopt the model developed in [1]. In [1], it is assumed that every source node has always packets to send; the channel conditions are assumed to be ideal; the collision probability is constant among all sources and independent of the past. Last, it is assumed that there is no limit on the number of retransmissions of a lost packet. In [1], the author derives the following expression for the transmission probability τ in terms of the protocol parameters

$$\tau = \frac{2(1-2p)}{(1-2p)(CW_0+1) + pCW_0(1-(2p)^m)} \tag{1}$$

where $m = \log_2(CW_{\max}/CW_0)$ and p denotes the collision probability seen by a transmitting source node. It is equal to $1-(1-\tau)^{N-1}$, where N denotes the number of active stations. For a given N, τ and p can be computed using a fixed-point approach. The author of [1] derives also an approximation of the optimal transmission probability, where the optimality refers to maximizing the system throughput. Let T_c denote the expected time taken by an unsuccessful transmission and σ denote the slot time length, the approximate optimal transmission probability is then given by

$$\tau^* = \frac{1}{N\sqrt{T_c/(2\sigma)}} \quad . \tag{2}$$

Insights on the Optimality of the Transmission Probability

The optimal transmission probability found in [1] will not only maximize the saturation throughput but will minimize as well the system response time. In other words, the idle time wasted over the wireless channel will be minimal. In this section, we will provide some insights on the optimality of the transmission probability.

The time over the wireless channel can be partitioned into successive *virtual transmission times*. A virtual transmission time initiates just after a successful transmission over the wireless channel, and ends at the end of the next successful transmission, as illustrated in Fig. 1. The wasted time in a virtual transmission time, denoted as $waste_\tau$, is due to collisions and idle slot times and can be written

$$waste_\tau = \mathbf{E}\left[N_cT_c + (N_c+1)\text{Idleness}\right] = \mathbf{E}[N_c]T_c + (\mathbf{E}[N_c]+1)\mathbf{E}[\text{Idleness}] \tag{3}$$

where $\mathbf{E}[N_c]$ and $\mathbf{E}[\text{Idleness}]$ respectively represent the average number of collisions and the average length of an idle period in a virtual transmission time (N_c and Idleness are statistically independent). In (3) it is assumed that all packets have the same size, hence T_c is a constant. The first and second terms of (3) respectively account for the total time wasted in collisions and the total idle time in a virtual transmission time. Observe that the virtual transmission time is simply $waste_\tau + T_s$ (T_s being the expected time taken by a successful transmission), so the normalized system throughput can be expressed as the ratio $\mathbf{E}[\text{payload}]/(waste_\tau + T_s)$. To derive expressions for $\mathbf{E}[N_c]$ and $\mathbf{E}[\text{Idleness}]$ we look at the distribution, in a virtual transmission time, of the number of collisions, N_c, and the idle period length. Let $P_I = (1-\tau)^N$, $P_s = N\tau(1-\tau)^{N-1}$ and $P_c = 1 - P_I - P_s$ be the probabilities of having an idle slot time, a successful transmission and a collision, respectively. Let σ be the slot time duration. We have

$$P[N_c = j] = \left(\tfrac{P_c}{P_c+P_s}\right)^j \tfrac{P_s}{P_c+P_s} \;, \qquad P[\text{Idleness} = j\sigma] = P_I^j(1-P_I)$$

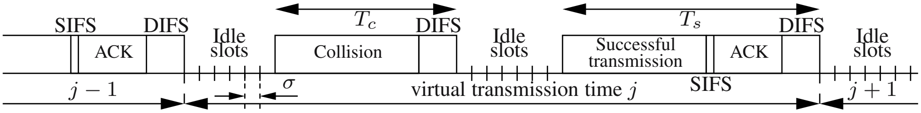


Fig. 1. An illustration of the virtual transmission time

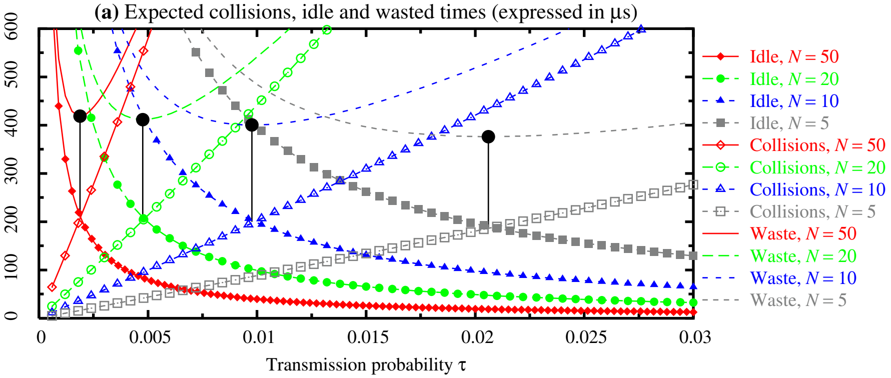


Fig. 2. Expected wasted time, expected time spent in collisions and expected idle time (all are in μs) in a virtual transmission time for different values of N ($T_c = 4335\mu$s and $\sigma = 20\mu$s).

for $j \in \mathbb{N}$, yielding $\mathbf{E}[N_c] = P_c/P_s$ and $\mathbf{E}[\text{Idleness}] = \sigma P_I/(1-P_I)$. After some calculus, it comes that the total time wasted in collisions and the total idle time are

$$\mathbf{E}[N_c]T_c = T_c\left(\frac{1-(1-\tau)^N}{N\tau(1-\tau)^{N-1}} - 1\right) \ , \qquad (\mathbf{E}[N_c]+1)\mathbf{E}[\text{Idleness}] = \frac{\sigma(1-\tau)^N}{N\tau(1-\tau)^{N-1}} \ .$$

It can easily be proved that $\mathbf{E}[N_c]T_c$ is a monotone increasing function of τ, whereas $(\mathbf{E}[N_c]+1)\mathbf{E}[\text{Idleness}]$ is a monotone decreasing function of τ. In Fig. 2, both terms are plotted against τ for different values of the number of contending stations N. Their sum, $waste_\tau$, is also plotted and its minimal value for each N has been marked. The minimal values of $waste_\tau$ naturally correspond to the optimal transmission probability for each N. As seen in Fig. 2, the minimal values of $waste_\tau$ correspond to the intersections of both collisions and idle times. In other words, the optimality is achieved when the wasted time is equally shared between collisions and idleness. This optimal operating point has been identified in [11, p. 243] for the CSMA slotted Aloha protocol and in [3] for the p-persistent IEEE 802.11 protocol.

Observe that having the system response time minimized at the optimal transmission probability does not guarantee that the packet delay is minimized as well. This minimization also relies on the protocol fairness and its ability to equally share the medium among contending sources.

4 Adaptive Binary Exponential Backoff (BEB)

According to equation (2), optimal system performance can be achieved in different network topologies by fine-tuning the transmission probability of the stations to the optimal transmission probability τ^* for each network topology. For a given network topology, transmitting at the optimal transmission probability can be achieved by well adjusting the size of the minimum starting contention window $CW_{\min}$. Note that a one-to-one relation between these two parameters can be obtained by inverting equation (1), and $CW_{\min}$ will then be expressed in terms of τ^* as follows:

Algorithm 1. Adjustment algorithm

```
Require: An estimation $\overline{N}$ of the number of active nodes
Ensure: Sub-optimal values of $CW_{\min}$ and $m$
1: Set $\tau = \tau^* = \frac{1}{\overline{N}\sqrt{T_c/2\sigma}}$
2: Set $p = 1 - (1 - \tau^*)^{\overline{N}-1}$
3: Compute $cw = \frac{(2-\tau^*)(1-2p)}{\tau^*(1-p-p(2p)^m)}$
4: Select $j$ such that $2^j CW_0$ is the closest to $cw$, $j \in [0, m]$
5: Set $CW_{\min} = 2^j CW_0$
6: Set $m = \log_2 \frac{CW_{\max}}{CW_{\min}}$ {$CW_{\max}$ is fixed}
```

$$CW_{\min} = \frac{(2 - \tau^*)(1 - 2p)}{\tau^*(1 - p - p(2p)^m)} \ . \tag{4}$$

Unfortunately, the parameter m will still depend on the previous value of $CW_{\min}$ as follows $m = \log_2(CW_{\max}/CW_{\min})$. However, this practically will not impact the new selected value of $CW_{\min}$ since the term $p(2p)^m$ can be neglected with respect to 1.

The operation of our algorithm can then be summarized as follows. After a failed transmission, the behavior is the same as in the standard. However, after a successful transmission, the initial window size of the binary exponential backoff mechanism is set to an optimal value, hereafter denoted $CW_{\min}$. The selection of this value is detailed in Algo. 1. Given an estimation of N, the approximate optimal transmission probability is computed using (2) (line 1), enabling the computation of the collision probability p (line 2). The optimal minimal size of the contention window size is computed according to (4) (line 3). Note that to be as close as possible to the standard, $CW_{\min}$ is imposed to take only those values defined by the standard $\{2^j CW_0, j = 0, \ldots, m\}$ (recall that CW_0 is the size of the minimal contention window in the standard) (lines 4–5). Last, the value of m is updated (line 6). Observe that the values of $CW_{\min}$ and m will be sub-optimal due to the fact that $CW_{\min}$ is made discrete.

The analytical throughput achieved by our adaptive algorithm, hereafter referred to as “Adaptive BEB”, can be obtained by substituting the new values of $CW_{\min}$ and m and the optimal value of p given in line 2 of Algo. 1 to compute the sub-optimal value of τ, and therefore the throughput achieved by Adaptive BEB will be $T = \mathbf{E}[\text{payload}]/(waste_\tau + T_s)$.

5 Estimation of the Number of Contending Stations

To adjust $CW_{\min}$, we need an estimation of N, hereafter denoted $\overline{N}$, that tracks its time-evolution. However, since the value of $CW_{\min}$ is made discrete, a reactive and adequately accurate estimator is sufficient. For instance, consider the case when $N = 30$. When applying Algo. 1, we would obtain the correct optimal value $CW_{\min} = 2^4 CW_0$ for $\overline{N} \in [20.45, 40.95]$.

As opposed to previously proposed methods (e.g. [2, 5, 10]) which estimate the current number of sources by measuring the channel activity, we propose to estimate the

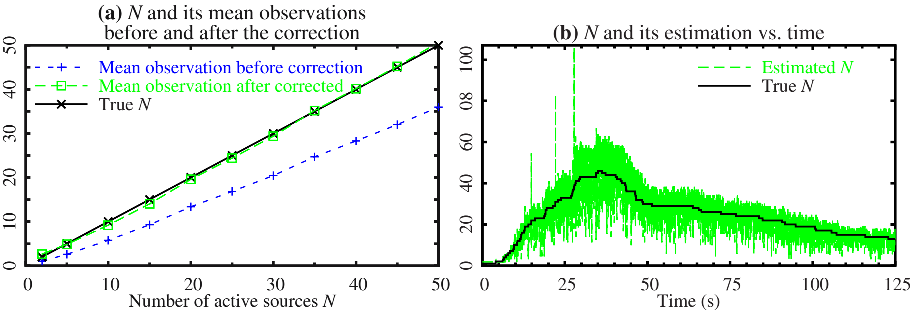


Fig. 3. (a) Mean observations of N vs. N. **(b)** Estimation of N vs. time in sample simulation

number of active stations directly by counting as active each station from which the concerned measuring station receives a *sign of life*, i.e. error-free data or RTS packets. The measurement period will be the virtual transmission time of the station, i.e. the time between its consecutive successful transmissions. The idea can be summarized as follows: since each active node is always filtering received packets whether they are destined to it or not, it can thus keep trace of a large number of active nodes in a given time interval by counting the number of *distinct* signs of life received in that time interval from these nodes. Let $\widehat{N}_n$ denote the count of distinct signs of life at the nth measurement period of a given station. For multiple reasons, these measurements cannot be used directly in Algo. 1 as an estimation of N. First, the count of signs of life cannot account for stations whose transmitted packets are corrupted, either by noise or collisions. Second, variable and relatively small measurement periods result in counting only a variable portion of all active stations. Therefore, based on the measurements $\{\widehat{N}_n\}_{n\in\mathbb{N}^*}$, an unbiased estimator should be devised. Observe that measurements collected over relatively large measurement times are more “accurate” than those collected over small measurement times, and should be preferentially treated. By observing that the expected length of a measurement period is reflected in cw, the value reached by the contention window at the end of the measurement time, one can use this value to proportionally weight the corresponding measurement, so that the following filter $\sum_{i=0}^{q-1} cw_{n-i}\widehat{N}_{n-i}/\sum_{i=0}^{q-1} cw_{n-i}$ can be used. Simulation analysis has shown that low values of the filter order q will suffice for a convenient performance, so hereafter, we set $q = 3$ so as to heighten reactiveness. However, and because of the fact that corrupted packets cannot contribute to the measurements, the previous ratio, henceforth referred to as the observation, underestimates N and needs thus to be corrected. To derive an appropriate correction on the observations, we have performed several simulations with different values of N. Figure 3(a) depicts the evolution of the observation against the actual number of sources. Obviously, a linear correction is needed, resulting in the following estimator

$$\overline{N}_n = a\left(\sum_{i=0}^{q-1} cw_{n-i}\widehat{N}_{n-i}\right) \Big/ \left(\sum_{i=0}^{q-1} cw_{n-i}\right) + b \tag{5}$$

with $a = 1.35405$, $b = 1.75998$ for $q = 3$. These latter values, which are independent of the nodes distribution, need to be adjusted in the case of noise over the channel; this issue is left for future work. Figure 3(b) illustrates the estimation of N over a simulation in which nodes arrivals are Poisson and activity time is exponentially distributed. The estimator exhibits good reactiveness to changes in N at the cost of large fluctuations, but these have only a limited impact on the performance of Algo. 1 because of the discretization process.

6 Enhancement of the Backoff Algorithm in Noisy Environments

The assumption of ideal channel conditions is in general unrealistic due to the existence of various "noise" factors (e.g. fading, shadowing, interference) that perturb the state of the wireless medium. Whenever a frame is noise-corrupted, both the standard and Adaptive BEB behave as if the loss is due to a collision, and the contention window of the backoff algorithm will be accordingly increased. Clearly, there is a flaw in the design of these algorithms due to the automatic invocation of the contention window increase process in the absence of CTS/ACK. Both algorithms lack to identify the cause of a missed CTS/ACK, and fail in adapting to error-prone environments. In the following, we present an optional extension that can be applied to Algo. 1 to enhance its performance in noisy environments. In this extension, we propose a Packet Error Rate (PER) estimator (Sect. 6.1), and a persistence mechanism that makes use of this estimator (Sect. 6.2). The resulting algorithm will be denoted as "Adaptive BEB^{++}".

6.1 PER Estimation

To estimate the Packet Error Rate, we propose to infer it from the measured corruption probability in a station virtual transmission time, and filter these inferred values through a simple Exponentially Weighted Moving Average (EWMA) filter. This method works well in both RTS/CTS and basic access modes. When a transmission occurs on the channel, all other stations within communication range receive the transmitted packet. Upon receiving a packet, every station verifies first its Check Redundancy Code (CRC) so that corrupted packets could be disregarded. Therefore, every station is capable of counting how many received packets are corrupted out of all of them. The ratio between the two counts, noted $\hat{p}_{cr}$, is nothing but a measure of the corruption probability, p_{cr}, perceived in a measurement time. The corruption probability can be written $p_{cr} = p_c + p_e(1 - p_c)$ where p_e is the PER and p_c is the collision probability seen on the channel by a measuring station. We have $p_c = 1 - (1 - \tau)^{N-1} - (N - 1)\tau(1 - \tau)^{N-2}$. Using $\overline{N}$ and τ^* we can infer p_c; the inferred value is denoted $\hat{p}_c$. Therefore, p_e can be inferred as follows $\hat{p}_e = (\hat{p}_{cr} - \hat{p}_c)/(1 - \hat{p}_c)$ and is used to feed the following EWMA filter: $\bar{p}_{e,n} = \alpha\bar{p}_{e,n-1} + (1 - \alpha)\hat{p}_{e,n}$, where $\bar{p}_{e,n}$ and $\hat{p}_{e,n}$ respectively denote the estimated and inferred PER in the nth measurement time at a given station. As for α, we have performed several simulations with $N = 25$ and an abruptly varying PER, as illustrated in Fig. 4(d). We have computed both expectation and variance of the error $p_e - \bar{p}_{e,n}$ for different values of α, and selected the value minimizing the mean error. This value is $\alpha = 0.95$ as can be seen from the table below. Observe in Fig. 4(d) how $\alpha = 0.95$ yields a highly reactive estimator.

α	0.9	0.95	0.99	0.995	0.999
$\mathbf{E}[p_e - \bar{p}_{e,n}]$	0.0174836	0.0153285	0.0192210	0.0259682	0.0714458
$\text{Var}[p_e - \bar{p}_{e,n}]$	0.0494257	0.0498566	0.0479849	0.0453496	0.0282030

6.2 The Persistence Algorithm

Unlike the standard and the Adaptive BEB mechanisms that upon a missed CTS/ACK blindly defer the transmission, we propose a persistence probability P_p such that the former behavior is undertaken only with probability $1 - P_\text{p}$. Upon a missed CTS/ACK and with probability P_p, the station retransmits the packet when the CTS/ACK timeout expires. In other words, P_p can be regarded as the probability of assuming that a failed transmission is due to a noise-corruption, not a collision. Obviously, the choice of this persistence probability is crucial for the algorithm to perform well. P_p should account for $\bar{p}_e$, the backoff stage, and the count of persistent trials within the current backoff stage. Clearly, it should increase with $\bar{p}_e$. Also, as a station increments its window size for a given packet retransmission, the collision probability decreases. If, however, the retransmission still fails, then it is more likely that it was due to a noise-corruption rather than a collision, and therefore, P_p should increase with the actual backoff stage that is equal to $\log_2(CW/CW_\text{min})$. Last, if, within the same stage, persistent retransmissions are repeatedly being unsuccessful, then it is more likely that the station is observing collisions and not corruption by the noisy channel, and as such, the persistence probability should decrease with $trials$, the count of persistent retransmissions within the same backoff stage. In our implementation, we have used the following expression of the persistence probability which exhibits all above-mentioned desired trends:

$$P_\text{p} = \bar{p}_e^{\frac{trials}{1+\log_2(CW/CW_\text{min})}} .$$

7 Simulation Results

In this section, we investigate the effectiveness of our proposal through simulations conducted in `ns-2` [12]. Simulated nodes are uniformly distributed in a 100m × 100m square and the power transmission is sufficiently high so that all the nodes are within communication range. All sources generate Constant Bit Rate (CBR) traffic. The protocol parameters are as follows: $CW_0 = 32$, $CW_\text{max} = 1024$ and the slot size is $\sigma = 20\mu$s. The expected collision duration T_c used in (2) corresponds to the maximum length of the data packet delivered by the MAC layer to the PHY layer and depends on the data rate. The proposed algorithms have been thoroughly tested considering both ad hoc and infrastructure modes of operation, fixed and abruptly changing number of sources as well as Poisson source arrivals. Both error-free and error-prone channels have been considered. Due to space limitations, we will discuss only 2 scenarios exhibiting the most relevant properties of the proposed algorithms.

Scenario A: N nodes are uploading CBR traffic to an access point. Starting with 10 active nodes for a duration of 25s, two 20-node bursts join the network at instants 25s

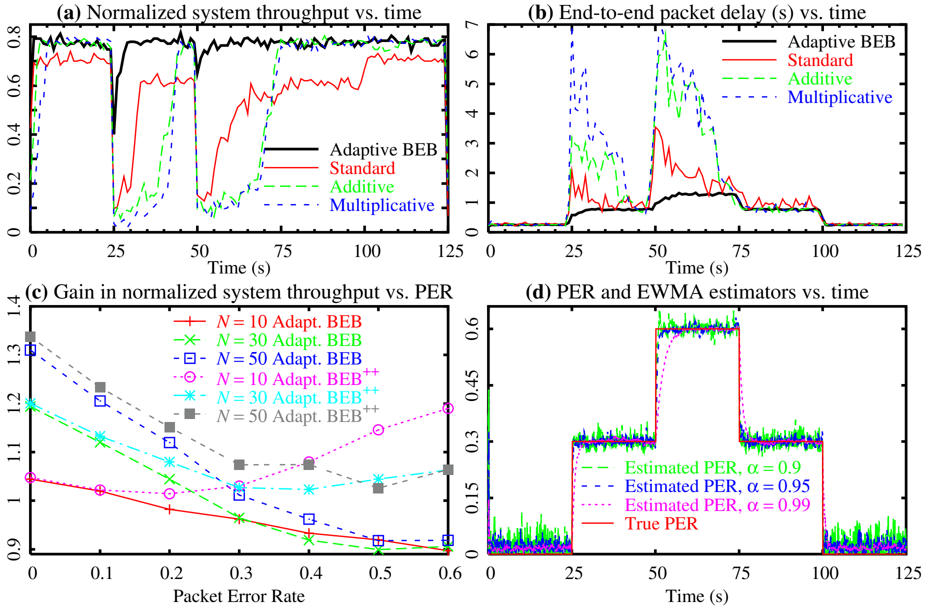


Fig. 4. (a) Simulative system throughput and **(b)** end-to-end delay (in s) over time (scenario A). **(c)** Throughput gain of Adaptive BEB and Adaptive BEB^{++} against the standard (scenario B). **(d)** PER and EWMA estimators vs. time for several α values.

and 50s, and then leave the network consecutively at instants 75s and 100s. Each node generates 1050B-packets every 5ms. Data rate at the physical layer is 2Mbps. There is no error on the wireless channel and the basic access mechanism is used. We have investigated the performance of 3 algorithms beside the standard: the Adaptive BEB, the additive increase/additive decrease algorithm [8] and the multiplicative slow decrease algorithm [7] referred to "Additive" and "Multiplicative" respectively. Figures 4(a) and 4(b) respectively illustrate the system throughput and the packet delay (in seconds) over time, when using the standard and the three adaptive algorithms. As can be seen, Adaptive BEB exhibits a steady performance over the simulation time whether in terms of throughput or packet delay, and is the only one to rapidly recover after the abrupt increase of N at times 25s and 50s. This can be explained by the fact that an optimal $CW_{\min}$ is rapidly selected, consequently avoiding the high number of collisions that are experienced by the standard. The slow decrease approaches [7, 8] perform very well when the number of sources is constant or changes smoothly. However, their performance degrades severely in the scenario at hand for 2 reasons. First, their control algorithms are relatively slow compared to the important change in the network state. Second, in the infrastructure mode of operation, some congestion occurs at the access point buffer yielding lost packets and subsequent missed ACKs. Buffer overflow is penalizing more the slow decrease approaches than the Adaptive BEB as in these approaches the contention window after a successful transmission will be unnecessarily large. In contrast, the Adaptive BEB controls the $CW_{\min}$ by using $\overline{N}$, so that only retransmissions affect the system performance.

Scenario B: There are $2N$ nodes in the network: N sources and N destinations operating in ad hoc mode with a data rate at the physical layer set to 11Mbps. The basic access mechanism is used. Each source generates 1000B-packets every 0.8ms. The error on the channel is modeled as a Bernoulli loss process. Packets are noise-corrupted with a probability PER set constant throughout each simulation runtime (100s). Two algorithms are investigated: the Adaptive BEB and the Adaptive BEB^{++}. Figure 4(c) plots, for different values of N, the throughput gain (defined as the ratio between both throughputs) achieved by each algorithm with respect to the standard versus the PER. Due to its persistence mechanism, Adaptive BEB^{++} shows a high robustness to error on the channel, and the gain obtained is always greater than 1 for different values of PER and N. For instance there is as much as 19% more throughput with Adaptive BEB^{++} when $N = 10$ and $\text{PER} = 0.6$. The Adaptive BEB has previously shown better performance than the standard. This is no longer true at high PER values. Actually, the Adaptive BEB is much more penalized by high PERs since in this case the average backoff time will be much higher than in the standard due to a larger starting contention window. There will be simply too much idle times over the channel.

8 Fairness Analysis

Fairness describes the MAC protocol capability to distribute the available resources equally among communicating terminals. There are a variety of fairness definitions intended to support different QoS and service differentiation. In our study, fairness is simply the equal partitioning of the bandwidth among all flows at hand. We have run simulations when either one of the following algorithms is used: Adaptive BEB or the standard when $\text{PER} = 0$, and Adaptive BEB^{++} or the standard when $\text{PER} = 0.4$. We considered 3 different values of N, namely, 10, 30 and 50. For each simulation, we have computed the index of fairness of Jain et al. [13] $f(x_1, x_2, \ldots, x_I) = \left(\sum_{i=1}^{I} x_i\right)^2 / \left(I \sum_{i=1}^{I} x_i^2\right)$ where x_i denotes user i allocation, and I the number of users sharing the bandwidth. This index returns the percentage of flows being treated fairly. To compute the index of fairness, we use the Sliding Window Method (SWM) introduced in [14]. A small sliding

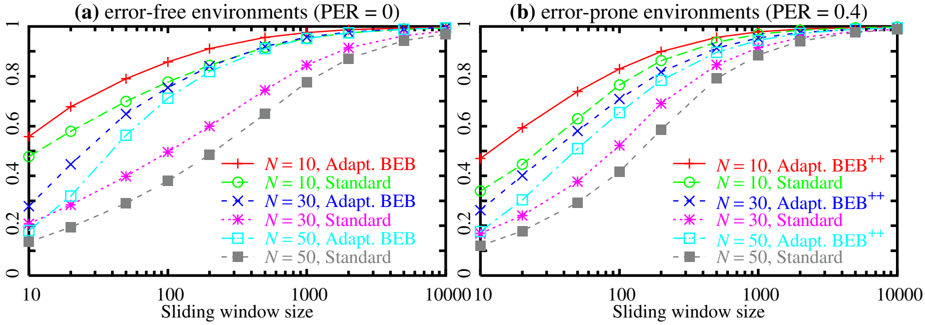


Fig. 5. Fairness index versus the sliding window size

window size allows the study of the short-term fairness whereas large sliding window sizes enable the long-term fairness one. The results are plotted in Fig. 5.

Observe that Adaptive BEB (respectively Adaptive BEB^{++}) achieves better fairness, i.e. higher index value, than the standard when $\mathrm{PER} = 0$ (respectively when $\mathrm{PER} = 0.4$), for any investigated value of N. The fact that our algorithm assigns to the contending nodes optimal and relatively equal $CW_{\min}$ allows on one hand to minimize the number of collisions, and consequently the risk of having largely different backoff intervals, and on the other hand, to obtain equal opportunities of accessing the channel for the various nodes.

9 Conclusion

In this paper, we have proposed and evaluated a simple, efficient and robust adaptive mechanism that can be easily incorporated within the IEEE 802.11 DCF function in order to optimize its performance. The proposed algorithm includes two mechanisms. The first optimally adjusts the minimum contention window size to the current network congestion level by using an on-line estimation of the number of active stations. The other mechanism extends the first to enhance its performance in the presence of noisy channels. Simulation results have shown that our proposed algorithm outperforms the standard in terms of throughput, packet delay, fairness and robustness to noise, for different scenarios and configurations. When compared to other proposals, our algorithm has shown better performance under several configurations.

Acknowledgements. The second author wishes to thank I. Aad and R. van der Mei for fruitful discussions at an early stage of this work.

References

1. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications **18**(3) (2000) 535–547
2. Cali, F., Conti, M., Gregori, E.: Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. IEEE/ACM Trans. on Networking **8**(6) (2000) 785–799
3. Cali, F., Conti, M., Gregori, E.: IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. IEEE J. on Sel. Areas in Comm. **18**(9) (2000) 1774–1786
4. IEEE Std. 802.11b, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. (1999)
5. Bianchi, G., Fratta, L., Oliveri, M.: Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In: Proc. of PIMRC '96. (1996)
6. Bononi, L., Conti, M., Gregori, E.: Runtime optimization of IEEE 802.11 wireless LANs performance. IEEE Transactions on Parallel and Distributed Systems **15**(1) (2004) 66–80
7. Aad, I., Ni, Q., Barakat, C., Turletti, T.: Enhancing IEEE 802.11 MAC in congested environments. In: Proc. of IEEE ASWN '04, Boston, Massachusetts. (2004)
8. Galtier, J.: Optimizing the IEEE 802.11b performance using slow congestion window decrease. In: Proc. of 16th ITC Specialist Seminar, Anvers, Belgique. (2004)
9. Nadeem, T., Agrawala, A.: IEEE 802.11 DCF enhancements for noisy environments. In: Proc. of PIMRC '04, Barcelona, Spain. (2004)

10. Bianchi, G., Tinnirello, I.: Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network. In: Proc. of IEEE INFOCOM '03. (2003)
11. Bertsekas, D., Gallager, R.: Data Network. Prentice Hall (1992)
12. The network simulator, version 2.28 (2005) `http://www.isi.edu/nsnam/ns/`.
13. Jain, R., Hawe, W., Chiu, D.: Quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC-TR-301, DEC (1984)
14. Koksal, C.E., Kassab, H., Balakrishnan, H.: An analysis of short-term fairness in wireless media access protocols (extended abstract). Perf. Eval. Rev., **28**(1) (2000) 118–119

A Comparison of Exact and ε-Approximation Algorithms for Constrained Routing

Fernando Kuipers[1], Ariel Orda[2], Danny Raz[2], and Piet Van Mieghem[1]

[1] Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands
{F.A.Kuipers, P.VanMieghem}@ewi.tudelft.nl
[2] Technion, Israel Institute of Technology, Haifa, Israel 32000
{ariel@ee, danny@cs}.technion.ac.il

Abstract. The Constrained Routing Problem is a multi-criteria optimization problem that captures the most important aspects of Quality of Service routing, and appears in many other practical problems. The problem is NP-hard, which causes exact solutions to require an intractable running time in the worst case. ε-approximation algorithms provide a guaranteed approximate solution for all inputs while incurring a tractable (i.e., polynomial) computation time. This paper presents a performance evaluation of these two types of algorithms. The main performance criteria are accuracy and speed.

Keywords: QoS routing, performance evaluation, RSP algorithms.

1 Introduction

One of the key issues in providing guaranteed Quality of Service (QoS) is *how to determine paths that satisfy QoS constraints*. Solving this problem is known as *Constrained routing* or *QoS routing*. The research community has extensively studied this problem, resulting in many QoS routing algorithms (see [5] for an overview and performance evaluation). Research has mainly focused on a two-parameter optimization problem called the Restricted Shortest Path (*RSP*) problem. Before presenting the formal definition of the RSP problem, we introduce some terminology and notation.

Let $G(N, L)$ denote a network topology, where $\{N\}$ is the set of N nodes and $\{L\}$ is the set of L links. The number of QoS measures (e.g., delay, hop count) is denoted by m. Each link is characterized by an m-dimensional link weight vector, consisting of m non-negative QoS weights ($w_i(u, v)$, $i = 1, ..., m$, $(u, v) \in \{L\}$) as components. The QoS measure of a path can be either *additive* (e.g., delay, jitter, the logarithm of packet loss), in which case the weight of a path equals the sum of the weights of its links, or *bottleneck* (e.g., available bandwidth), in which case the weight of a path is the minimum (or maximum) of the weights of its links. Without loss of generality [9], we assume all QoS measures to be additive.

The RSP problem is formally defined as follows.

Definition 1. *Restricted Shortest Path (RSP) problem*: Consider a network $G(N, L)$. Each link $(u, v) \in \{L\}$ is specified by $m = 2$ nonnegative measures:

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 197–208, 2006.

a *cost* $c(u,v)$ and a *delay* $d(u,v)$. Given a *delay constraint* Δ, the RSP problem consists of finding a path P^* from a source node s to a destination node d such that $d(P^*) \leq \Delta$ and $c(P^*) \leq c(P)\ \forall P : d(P) \leq \Delta$, where $c(P) \overset{\text{def}}{=} \sum_{(u,v)\in P} c(u,v)$ and $d(P) \overset{\text{def}}{=} \sum_{(u,v)\in P} d(u,v)$.

The RSP problem is known to be NP-hard [1]. To cope with this worst-case intractability, heuristics and ε-approximations have been proposed, as well as a few exact algorithms.

As described in [5], many studies focused on heuristic solutions, which may perform well in certain scenarios. However, in the most general case they cannot provide any performance guarantee, which makes them unpredictable. We focus on the two classes of exact and ε-approximation algorithms, which can (rigorously) provide a predefined level of QoS guarantees. For the ε-approximation algorithms mainly theoretical results exist and no empirical results are published. Exact algorithms provide the optimal solution, however their running time may be very high in the worst case. In this paper we evaluate two representative algorithms, distinguish their worst cases, provide empirical results and discuss and compare the relative strengths of the two approaches.

The outline of the paper is as follows. In Section 2 we describe the two algorithms: we choose SAMCRA [9] as a representative of the class of exact RSP algorithms and SEA [6] as a representative of the class of RSP ε-approximation algorithms. In Section 3 we delineate the worst-case scenarios of each of the two algorithms. In Section 4 we conduct an empirical comparison between the two algorithms. Finally, we discuss some open problems in Section 5 and provide a brief conclusion in Section 6.

2 RSP Algorithms

2.1 SAMCRA

SAMCRA [9] stands for Self-Adaptive Multiple Constraints Routing Algorithm and is a general exact QoS algorithm, which incorporates four fundamental concepts: (1) a nonlinear measure for the path length. When minimizing a linear function of the weights, solutions outside the constraints area may be returned. An important corollary of a nonlinear path length is that *the subsections of shortest paths in multiple dimensions are not necessarily shortest paths themselves.* This necessitates to consider in the computation more paths than only the shortest one, leading to (2) a k-shortest path approach. The k-shortest path algorithm is essentially Dijkstra's algorithm that does not stop when the destination is reached, but continues until the destination has been reached by k different paths, which succeed each other in length. To reduce the search space we use (3) the principle of non-dominated paths[1], and (4) the look-ahead concept. The latter precomputes (via Dijkstra's algorithm) one or multiple shortest

[1] Often also referred to as Pareto optimality. A path P is dominated by a path Q if $w_i(Q) \leq w_i(P)$, for $i = 1, ...m$, with inequality for at least one i.

path trees rooted at the destination and then uses this information to compute end-to-end lower bounds to reduce the search space. SAMCRA can be used with different length functions, and can therefore be easily adapted to solve the RSP problem. The nonlinear length that we have used is:

$$l(P) = \begin{cases} c(P), & \text{if } d(P) \leq \Delta \\ \infty, & \text{else} \end{cases} \tag{1}$$

By employing this length function, SAMCRA can guarantee to find the minimum-cost path within the delay constraint.

2.2 SEA

SEA [6] stands for Simple Efficient Approximation and is an ε-approximation algorithm that (like most ε-approximation algorithms) specifically targets the RSP problem. ε-approximation algorithms are characterized by a polynomial complexity and ε-optimal performance. An algorithm is said to be ε-optimal if it returns a path whose cost is at most $(1+\varepsilon)$ times the optimal value, where $\varepsilon > 0$ and the delay constraint is strictly obeyed. ε-approximation algorithms perform better in minimizing the cost of a returned feasible path as ε goes to zero. However, the computational complexity is proportional to $1/\varepsilon$, making these algorithms impractical for very small values of ε. SEA is based on Hassin's algorithm [3], which has a complexity of $O((\frac{LN}{\varepsilon} + 1)\log\log B)$, where B is an upper bound on the cost of a path. It is assumed that the link weights are positive integers. This ε-approximation algorithm initially determines an upper bound (UB) and a lower bound (LB) on the optimal cost. For this, the algorithm initially starts with $LB = 1$ and $UB =$ sum of (N-1) largest link-costs, and then systematically adjusts them using a *testing* procedure. Once suitable bounds are found, the approximation algorithm bounds the cost of each link by rounding and scaling it according to: $c'(u,v) = \left\lfloor \frac{c(u,v)(N+1)}{\varepsilon LB} \right\rfloor + 1 \;\forall\; (u,v) \in \{L\}$. Finally, it applies a pseudo-polynomial-time algorithm on these modified weights. SEA improves upon Hassin's algorithm by finding better upper and lower bounds and by improving the testing procedure. In this way SEA obtains the polynomial complexity of $O(LN(\log\log N + \frac{1}{\varepsilon}))$.

It is also worth mentioning that there is another class of approximation algorithms, e.g. [2], that approximate the delay constraint rather than the cost. Indeed, this is a heavier compromise, but the reward is in terms of a smaller running time. Yet another approach is to specialize on the network topology (e.g., assume a hierarchical structure) and thus provide an exact and computationally tractable solution [7].

3 Worst-Case Scenarios

NP-hard problems may be solvable in some (or even many) instances, while displaying intractability in the worst case. It is therefore important to gain some understanding at what constitutes a worst-case scenario for a particular problem or algorithm.

3.1 Exact Algorithms

Worst-case scenarios for exact QoS algorithms were identified in [4], and according to [5] they also resulted to be worst-case scenarios for several heuristics. Summarizing [4], the intractability of the constrained routing problem hinges on four factors, namely: (1) The underlying topology, because the number of paths in some classes of topologies can be bounded by a polynomial function of N; based on empirical results [4], other classes of topologies, like the class of random graphs that have a small expected hop count, also appear to be computationally solvable. (2) Link weights that can grow arbitrarily large or have an infinite granularity; when link weights are bounded and have a finite granularity, which is often the case in practice, it can be proved that the constrained routing problem is solvable in polynomial time; in fact, this is the property that ε-approximation algorithms rely on to guarantee a polynomial complexity. (3) A very negative correlation among the link weights; empirical results [4] indicate that there is hardly any "intractability" for the entire range of correlation coefficients $\rho \in [-1, 1]$, except for extreme negative values. (4) The values of the constraints: if they are very large, then it is easy to find a path within the constraints, while if they are very small, then it is easy to verify that there is no path that meets the constraints. If, indeed, the four above-mentioned conditions are all necessary to "induce intractability," they could allow network and service providers to properly dimension their infrastructures so as to avoid intractable scenarios.

3.2 ε-Approximation Algorithms

The class of ε-approximation algorithms are based on entirely different concepts and may not be affected by the worst-case scenarios of exact algorithms. In this section we delineate the worst-case scenarios for ε-approximation algorithms, and in particular for SEA.

The rounding and scaling performed by SEA prevents that a solution that is exactly a factor $(1+\varepsilon)$ larger than optimal can be returned. The scaled weights are computed via $c'(u,v) = \left\lfloor \frac{c(u,v)(N+1)}{\varepsilon LB} \right\rfloor + 1 \ \forall \ (u,v) \in \{L\}$ and hence we have that $c(u,v) \leq \frac{c'(u,v)\varepsilon LB}{N+1} \leq c(u,v) + \frac{\varepsilon LB}{N+1}$. The maximum error that can be made along any path therefore equals $\frac{(N-1)\varepsilon LB}{N+1} \leq \frac{(N-1)\varepsilon c(P^*)}{N+1} < \varepsilon c(P^*)$. The maximum path error of $\varepsilon c(P^*)$ can only be approximated from below for large N.

The factor that affects performance is not so much the topology as the distribution of weights over the links. Let us consider two nodes, s and d, interconnected by two links as displayed in Figure 1. Let the delay of each link be 1, and let the costs be $c(l_1) = 1$, $c(l_2) = 1 + \frac{\varepsilon LB}{N+2}$. We assume that $\frac{N+1}{\varepsilon LB}$ is an integer number, then scaling the link costs results in $c'(l_1) = c'(l_2) = \frac{N+1}{\varepsilon LB} + 1$. Hence, due to the scaling performed by the algorithm, the weights of the two links would appear identical, and the algorithm may pick link l_2, which is a factor $(1 + \frac{\varepsilon}{4})$ more costly than link l_1, when $LB = c(P^*) = 1$. SEA cannot return

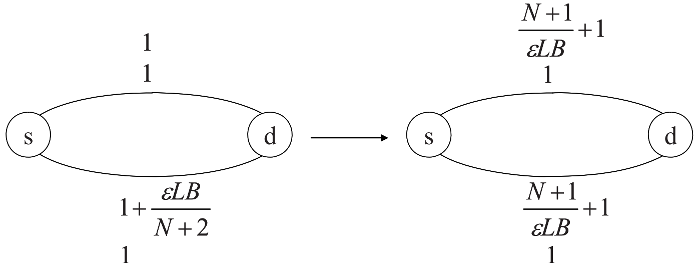


Fig. 1. Example topology consisting of two nodes and two links, where each link is characterized by a cost and a delay. The left topology represents the original weights, while the right topology gives the scaled weights (according to SEA).

a path that is a factor $(1 + \frac{\varepsilon}{3})$ more costly than optimal in this topology. Note that, depending on the implementation details of the algorithm, either of the two paths could be chosen. This source of "randomness" reduces the expected error over multiple graphs.

Another measure that determines the worst-case error of SEA is the value of the lower bound LB. SEA first determines upper and lower bounds, such that $\frac{UB}{LB} \leq N$. Hence, for the lower bound holds that $\frac{c(P^*)}{N} \leq LB \leq c(P^*)$. In case $LB = \frac{c(P^*)}{N}$, the worst-case error that SEA could make is upper bounded by $\frac{\varepsilon c(P^*)}{N}$.

Finally, in a general topology, the weights are unlikely to constitute worst-case errors. To obtain a worst-case error, the link weights should be chosen from two classes, namely link weights that, when scaled and rounded, do not lead to an error and link weights that, when scaled and rounded, give the maximum attainable error. The optimal path would then consist of the "error-free" link weights, while the approximation algorithm could return in the worst case a path that includes only the "erroneous" link weights. If the weights are randomly assigned to the links, then there is a smoothing effect over the various links. So, for pushing the algorithm to its limit, one could (either or both):

- Consider very simple topologies, with a small number of edges and low connectivity.
- Assume some correlation among the weights of consecutive links, in an attempt to cancel the "smoothing effect." In addition, the weights of the links should be chosen out of a small set, in which the differences are such that the scaling operation would incur the maximal possible error.
- We should focus on large values of the weights and the delay constraint, since for small values a pseudo-polynomial algorithm would provide a solution that is both optimal and computationally solvable.

4 Performance Evaluation

We have performed a comprehensive set of simulations to compare between SAMCRA and SEA. We have used Waxman graphs [8], complete graphs, random graphs of the type $G_p(N)$, where p is the link density, power-law graphs, and lattices. In each class of graphs, the delay and cost of every link $(u, v) \in \{L\}$ were taken as independent uniformly distributed random integers in the range $[1, M]$. However, for the class of lattices, the delay and the cost of every link (u, v) were also negatively correlated: the delay was chosen uniformly from the range $[1, M]$ and the corresponding cost was set to $M + 1$ minus the delay. Simulations for different values of M did not display any significant differences, so we have chosen $M = 10^5$. In each simulation experiment, we generated 10^4 graphs and selected nodes 1 and N as the source and destination, respectively. For lattices, this corresponds to a source in the upper left corner and a destination in the lower right corner, leading to the largest minimum hop count. For power-law graphs, this corresponds to a source that has the highest nodal degree and a destination that has the lowest nodal degree in the graph. For the other classes of graphs, this is equivalent to choosing two random nodes.

The delay constraint $\varDelta$ was selected as follows. First, we computed the least-delay path (LDP) and the least-cost path (LCP) between the source and the destination using Dijkstra's algorithm. If the delay constraint $\varDelta < d(\text{LDP})$, then there is no feasible path. If $d(\text{LCP}) \leq \varDelta$, then the LCP is the optimal path. Since these two cases are easy to deal with, we compared between the algorithms considering the values $d(\text{LDP}) < \varDelta < d(\text{LCP})$, as follows:

$$\varDelta = d(\text{LDP}) + \frac{x}{4}(d(\text{LCP}) - d(\text{LDP})) \tag{2}$$

In all simulations we chose $x = 2$, except when evaluating the influence of the constraints, in which case we considered $x = 0, 1, 2, 3, 4$.

4.1 Simulation Results

SAMCRA always finds the optimal path within the delay constraint. We therefore evaluated SEA based on how successful it is in minimizing the cost of a returned feasible path, when compared to SAMCRA. The effective approximation α of SEA is defined as

$$\alpha = \frac{c(P_{SEA})}{c(P_{SAMCRA})} - 1$$

where $c(P_x)$ is the cost of the feasible paths that are returned by algorithm x. We plot $E[\alpha]$, $var[\alpha]$, and $\max[\alpha]$ based on the 10^4 iterations. We also report the *execution time* of the compared algorithms. Figure 2 displays the effective approximation α and execution time as a function of ε for lattice graphs with $N = 100$, and independent uniformly distributed random link weights.

We can clearly see that $\alpha << \varepsilon$, which means that SEA hardly or never reaches a worst-case performance. Even the performance for $\varepsilon = 1$ is surprisingly good. The reason that $\alpha << \varepsilon$ is partly due to the assignment of the link weights according to a uniform distribution. Given that the link costs are uniformly distributed in the range $[1, M]$, then the scaled and rounded costs are approximately uniformly distributed in the range $[\left\lfloor \frac{(N+1)}{\varepsilon LB} \right\rfloor + 1, \left\lfloor \frac{M(N+1)}{\varepsilon LB} \right\rfloor + 1]$.

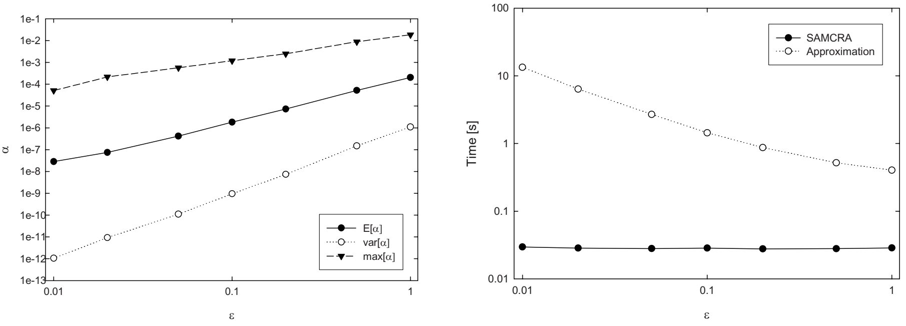


Fig. 2. Effective approximation α and execution time as a function of ε. The results are for Lattice graphs with $N = 100$, and the link weights are independent and uniformly distributed random variables.

As any real number x can be written as $x = \lfloor x \rfloor + \langle x \rangle$, where $\lfloor x \rfloor$ denotes the largest integer smaller or equal to x and where $\langle x \rangle \in [0, 1)$ denotes the fractional part of x, the round-off error of link (u, v) equals $1 - \left\langle \frac{c(u,v)(N+1)}{\varepsilon LB} \right\rangle$, for which holds $0 \leq 1 - \left\langle \frac{c(u,v)(N+1)}{\varepsilon LB} \right\rangle \leq 1$. Assuming that $\frac{(N+1)}{\varepsilon LB}$ is a fixed fractional number that is known to SEA before it executes its main procedure, the size of the round-off error is determined by the costs $c(u, v)$. Since these costs are uniformly distributed, we believe that the round-off errors are well approximated by a uniform distribution. If this holds, then our expected round-off error on a link is only half its worst-case value.

The expected α displays an approximately linear increase on the log-log scale, with a slope that is almost equal to 2. Therefore, in our simulated range, changing the value of ε has a quadratic impact on the effective approximation α. We can also see a clear correspondence between ε and the execution time: the larger ε, the smaller the execution time. The results approximately follow a linear line with a slope of -1 on a log-log scale, which indicates that the time is inversely proportional to ε, as was expected from the worst-case time complexity $O(LN(\log\log N + \frac{1}{\varepsilon}))$. However, even for $\varepsilon = 1$ the execution time of SEA is still by an order of magnitude larger than the execution time of SAMCRA. Figure 3 plots the effective approximation α as a function of the constraint values. A larger constraint means that more paths obey it. This larger search space results in a higher probability of making an erroneous decision (within the ε margin). The execution times of SAMCRA and SEA seem hardly influenced by the different constraints. Actually, by choosing x in Equation (2) as $x = 0$ or $x = 4$, the RSP problem is polynomially solvable, with solutions LDP and LCP respectively. For $x = 1, 2, 3$ SAMCRA is able to solve the RSP problem in a similar time span, suggesting that these simulated instances were also polynomially solvable.

Figure 4 displays the effective approximation α and execution time as a function of N.

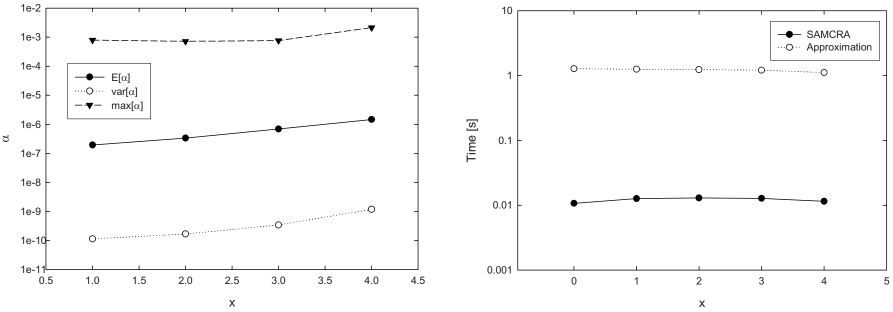


Fig. 3. Effective approximation α and execution time as a function of x in equation (2). The results are for Lattice graphs with $\varepsilon = 0.1$ and $N = 100$, and the link weights are independent and uniformly distributed random variables.

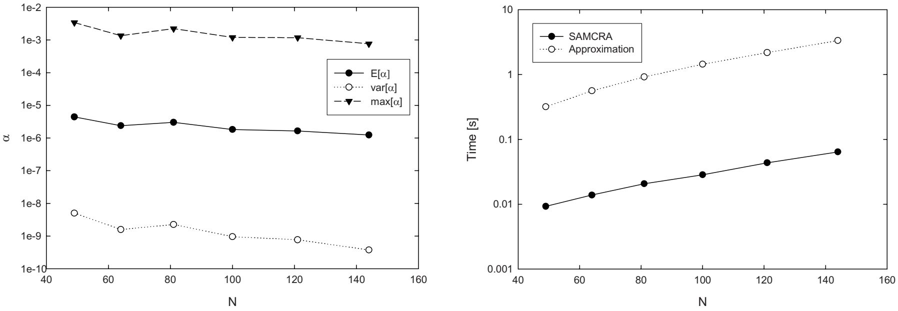


Fig. 4. Effective approximation α and execution time as a function of N with $\varepsilon = 0.1$. The results are for lattice graphs, and the link weights are independent and uniformly distributed random variables.

We can see that α slightly decreases with N. If N grows, there may be many paths that have a length close to the shortest feasible path. Finding one of these paths is less difficult than finding the true RSP path. The relative difference in time between SAMCRA and SEA remains fairly constant: SAMCRA is more than 10 times faster than SEA.

Figure 5 displays the results for negatively correlated random link weights. According to [4], this simulation setting corresponds to a worst-case scenario for exact algorithms.

Contrary to the decrease of α in Figure 4, we observe an increase of $E[\alpha]$ with N. Also, the values of α are much higher (considering the smaller values of N). Therefore, this worst-case scenario for exact algorithms also seems to affect ε-approximation algorithms, although not to the extent of constituting a worst-case scenario for SEA. The difference in execution time is clear: SAMCRA incurs an exponential computation time, whereas SEA is (always) a polynomial-time algorithm. Therefore, there is a cross-over point (at $N = 40$), where SAMCRA starts to run slower than SEA.

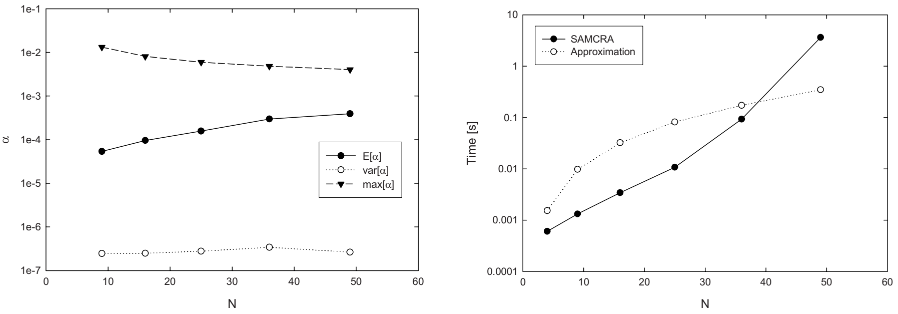


Fig. 5. Effective approximation α and execution time as a function of N with $\varepsilon = 0.1$. The results are for Lattice graphs, and the link weights are negatively correlated, uniformly distributed random variables.

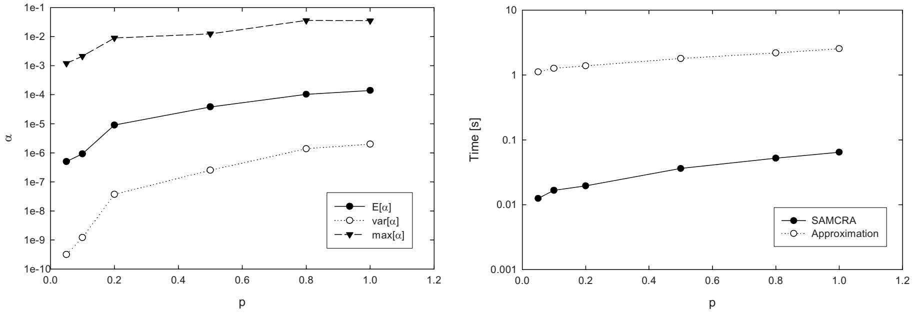


Fig. 6. Effective approximation α and execution time as a function of the link density p. The results are for random graphs with $\varepsilon = 0.1$, $N = 100$, and the link weights are independent and uniformly distributed random variables.

We have simulated in the class of random graphs with different link densities p ($p = 1$ corresponds to the class of complete graphs).

The values of α in Figure 6 increase with p, which suggests that SEA has more difficulty with dense graphs. Dense graphs have more links than sparse graphs and hence the probability of making round-off errors increases. Also, the denser a graph becomes, the shorter the expected hop count will be. With a short expected hop count, situations like in Figure 1 are more likely to occur than when the expected hop count is large, like in the class of lattices. A small effective approximation was also observed for the sparse Waxman graphs. The effective approximation α and execution time, as function of ε and N, in the class of Waxman graphs displayed a similar trend as in Figure 2 for the class of lattices, and hence are not plotted here.

We have also simulated in the class of power-law graphs, which are considered to contain the Internet graph. In power-law graphs the nodal degree distribution

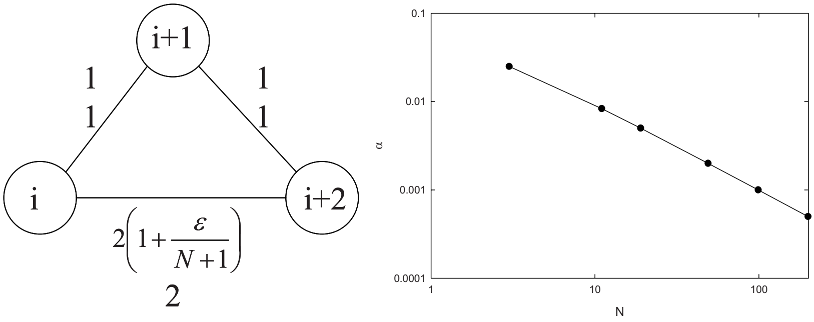


Fig. 7. Effective approximation α as a function of N. The results are for the chain topology (on the left, $i = 1, ..., N-2$), with $\varepsilon = 0.1$.

is $\Pr[d = i] = ci^{-\tau}$, where c is a constant such that $\sum_{i=1}^{N-1} ci^{-\tau} = 1$. Measurements in the Internet suggest that $\tau \approx 2.4$ and therefore we have chosen this value for the generation of our power-law graphs. Since the source referred to the node with the highest degree and the destination to the node with the lowest degree, the probability that there is only one path between source and destination is much higher in this class of power-law graphs than in the other considered classes of graphs. Our simulations for different ε showed that for $N = 100$ and $\varepsilon < 0.1$, α was zero. Furthermore, we deduced that the effective approximation α in the class of power-law graphs with $\tau \approx 2.4$ was the lowest among the considered classes of graphs.

Finally, we have simulated with a chain topology. By choosing the weights as in Figure 7, the error when rounding and scaling link $(i, i+2)$ equals $\frac{2\varepsilon}{N+1}$ and the total error that can be accumulated in the worst case along the lower path with $\frac{N-1}{2}$ hops is $\frac{N-1}{N+1}\varepsilon$. Since the optimal cost equals $N-1$, the effective approximation α can be found to obey $\alpha = \frac{(N-1)\varepsilon}{(N-1)(N+1)} = \frac{\varepsilon}{N+1}$, which perfectly matches our result in Figure 7, as seen by the straight line on a log-log scale.

4.2 Simulation Conclusions

In this subsection we summarize the conclusions that can be drawn from our simulation results.

1. Besides the better performance, the running time of the exact algorithm SAMCRA was at least ten times faster than the running time of SEA, in all simulated scenarios except for the constructed worst-case scenario of Figure 5.
2. The actual performance of SEA, as measured by the effective approximation α, was much better than the theoretical $(1+\varepsilon)$ upper bound.
3. The combination of many paths with a small hop count between source and destination leads to larger α on average than in the case of a large hop count or very few paths.

4. Changing the value of ε seems to have a quadratic impact on the effective approximation α.
5. The correspondence between the time t that SEA needs to solve an instance of the RSP problem and ε, nicely follows $t \sim \frac{1}{\varepsilon}$.

5 Discussion

SEA has a much better performance than the theoretical $(1 + \varepsilon)$ bound. The question therefore rises if we can make this bound sharper without increasing the time complexity. For instance, instead of only rounding up, one could consider rounding to the nearest number (e.g., if the granularity is 0.1 then $0.57 \rightarrow 0.6$ and $0.52 \rightarrow 0.5$). The overall worst-case error ε will then be halved and the expected error might tend to 0 (under a uniform distribution of the link weights).

The extension of RSP approximation algorithms like SEA to the more general QoS algorithms that handle $m > 2$ constraints would still be polynomial. However, the complexity would increase with $O(N^m)$, which may be prohibitive.

It is possible to devise approximation schemes that can also work with real weights. This can be done via an extra phase of rounding and scaling. The solution will still be polynomial, but a second source of inaccuracy (that can also be bounded) is introduced.

How to take advantage of the strengths of SAMCRA and SEA? One approach is to invoke SAMCRA with a running time "budget" T, within which it attempts to retrieve the optimal solution. In case SAMCRA encounters a hard instance, T may not suffice to accomplish this task. In this case, SAMCRA is halted and the SEA algorithm is invoked. The combined approach has the following properties: it guarantees an ε-optimal solution, it has a polynomial worst-case running time, and empirical evidence shows that, usually, an optimal solution would be found quickly.

6 Conclusions

The Restricted Shortest Path (RSP) problem seeks to minimize the cost of a path while obeying a delay constraint. The importance of this problem is undisputed, since it appears in many different research fields and plays a key role in Quality of Service (QoS) routing. Unfortunately, the RSP problem is NP-hard. Many algorithms have been proposed, which can be subdivided into the classes of exact solutions, ε-approximations, and heuristics. Only the first two classes can provide some (rigorous) level of guarantee on the optimality of the solution. We have therefore focused on these two classes, represented by the exact SAMCRA algorithm and the ε-approximation algorithm SEA. ε-approximation algorithms mainly have been studied theoretically, providing worst-case bounds, but not empirically. We have therefore compared SEA to SAMCRA. In worst-case scenarios, the complexity of SAMCRA is prohibitively high, but in most instances it ran significantly faster than SEA. SEA, on the other hand has a very good accuracy and polynomial running time.

References

1. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.
2. A. Goel, K.G. Ramakrishnan, D. Kataria, D. Logothetis, "Efficient Computation of Delay-sensitive Routes from One Source to All Destinations," Proc. of IEEE INFOCOM, pp. 854-858, 2001.
3. R. Hassin, "Approximation schemes for the restricted shortest path problem," Mathematics of Operations Research, vol. 17, no. 1, pp. 36-42, February 1992.
4. F.A. Kuipers and P. Van Mieghem, "The impact of correlated link weights on QoS routing," Proc. of the IEEE INFOCOM Conference, vol. 2, pp. 1425-1434, April 2003.
5. F.A. Kuipers, T. Korkmaz, M. Krunz and P. Van Mieghem, "Performance Evaluation of Constraint-Based Path Selection Algorithms," IEEE Network, vol. 18, no. 5, pp. 16-23, September/October 2004.
6. D.H. Lorenz and D. Raz, "A simple efficient approximation scheme for the restricted shortest path problem," Operations Research Letter, vol. 28, no. 5, pp. 213-219, June 2001.
7. A. Orda, "Routing with end-to-end QoS guarantees in broadband networks," IEEE/ACM Transactions on Networking, vol. 7, no. 3, pp. 365-374, 1999.
8. P. Van Mieghem, "Paths in the simple random graph and the waxman graph," Probability in the Engineering and Informational Sciences (PEIS), no. 15, pp. 535-555, 2001.
9. P. Van Mieghem and F.A. Kuipers, "Concepts of exact quality of service algorithms," IEEE/ACM Transactions on Networking, vol. 12, no. 5, pp. 851-864, October 2004.

Path Selection Techniques to Establish Constrained Interdomain MPLS LSPs*

Cristel Pelsser and Olivier Bonaventure

CSE Department, Université catholique de Louvain, Belgium
{pelsser, bonaventure}@info.ucl.ac.be

Abstract. MultiProtocol Label Switching (MPLS) is used today inside most large Service Provider (SP) networks. In this paper, we analyze the establishment of interdomain MPLS LSPs with QoS constraints. These LSPs cross diverse SP networks that may belong to different companies. We show that using the standard BGP route for the establishment of such LSPs is not sufficient. We propose two path establishment techniques that rely on RSVP-TE and make use of Path Computation Elements (PCEs). Our simulations show that these techniques increase the number of constrained MPLS LSPs that can be established across domain boundaries.

1 Introduction

During the last years, MultiProtocol Label Switching (MPLS) has been deployed by most large SP networks. Initially, MPLS was offered as a replacement for ATM. However, the main driver for the current deployment of MPLS is its ability to provide new services with stringent Service Level Agreements (SLAs) such as layer-2 and layer-3 Virtual Private Networks (VPNs) as well as Voice and Video over IP. Most of these services are already deployed inside single SP networks. However, customers now require world-wide VPN and VoIP services. Therefore, SPs need to collaborate to offer these services across multiple SP networks.

Inside a single SP network, the provision of MPLS-based services with stringent bandwidth and delay requirements is typically achieved by using the Traffic Engineering (TE) extensions to the ISIS/OSPF routing protocol. These extensions enable to distribute with ISIS/OSPF the link loads and delays. Based on this information, each Label Switching Router (LSR) can use a Constrained Shortest Path First (CSPF) algorithm to find a constrained path toward any router inside the SP network. Then, it can use the Resource reSerVation Protocol with Traffic Engineering extensions (RSVP-TE) to signal the establishment of a traffic engineered MPLS Label Switched Paths (LSP) along this path. However, when traffic engineered LSPs with QoS and delay constraints must be terminated at a router in another SP network the selection of the path becomes a problem [8]. The CSPF algorithm cannot be used to find a constrained path between

* This work was partially funded by the Walloon Government (DGTRE) in the framework of the TOTEM project (http://totem.info.ucl.ac.be) and supported by the E-NEXT NoE funded by the European Commission.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 209–220, 2006.

two LSRs in different interconnected SP networks anymore. This is because the networks exchange routing information by using the Border Gateway Protocol (BGP). In contrast to OSPF-TE/ISIS-TE, BGP only provides reachability information. It does not distribute complete topology, delay and bandwidth information.

In this paper, we evaluate techniques that allow to establish traffic engineered or constrained LSPs across multiple SP networks. Our paper is organized as follows. In section 2, we introduce the issues that arise when considering TE across domain boundaries. Then, we present, in section 3, the path selection techniques that we evaluate in this paper. We propose two heuristics for the selection of the ingress node in the downstream domains and combine them with one of the techniques. Next, we evaluate the path selection techniques in section 4. Finally, we conclude the paper.

2 Interdomain Issues

BGP is the routing protocol used between SP networks, also called Autonomous Systems (ASs). As we have already mentioned, BGP only provides reachability information for the destinations. More precisely, it only provides the addresses of Next Hops (NHs), the nodes at the border of the domain, that are able to forward the packets to a given destination. The QoS properties of the paths, such as the delay and bandwidth, behind these NHs are not provided. This results in several limitations for the computation and establishment of constrained interdomain LSPs.

Firstly, inside an AS[1], all routers learn the complete topology of the AS by means of ISIS/OSPF. Thus, each router is able to compute the complete path from head-end to tail-end node for an LSP contained in the AS. However, the topology of an AS is hidden to routers outside the AS, for confidentiality purposes [10]. As a consequence, a single node is not able to compute the end-to-end path for an LSP crossing multiple ASs. Therefore, the computation of such a path has to be distributed among multiple nodes, where each node computes a segment of the path based on its knowledge of the local AS topology and the interdomain reachability information provided by BGP.

Secondly, we have shown in [8] that a router only possesses a subset of the possible routes for a destination. Moreover, the set of routes learned by a BGP router are not necessarily the best possible routes with regard to the end-to-end delay and the available bandwidth. The BGP routes are first selected based on local preferences and the AS path length. However, Huffaker et al. have shown in [7] that the AS path length does not reflect the delay of the path. Thus, interdomain routes with a low delay may never be learned by some routers. The diversity of the BGP routes available at each router is not sufficient to successfully compute constrained interdomain LSPs.

Extensions to BGP in order to advertise the QoS of the interdomain routes are proposed in [1]. However, such extensions have not been evaluated nor deployed. In [13] and [6], the authors define an architecture with a centralized entity inside each domain. They propose to define a new interdomain routing protocol to be used between the entities and to exchange QoS information with this routing protocol. Up to now such a routing protocol has not been defined. It is not currently possible to know a priori the

[1] We consider ASs composed of a single IGP area. This is the most common deployment today.

QoS that can be provided along an interdomain route. Thus, in this paper we rely on heuristics to estimate the QoS of a route.

3 Path Selection Techniques

In this section we present four path computation techniques for constrained interdomain MPLS LSPs. The last two techniques are based on the same principle, ERO expansion. However, they make use of two different heuristics that are proposed in this section.

3.1 Standard IP forwarding

The simplest technique to establish an interdomain MPLS LSP is to follow the same path as the normal IP packets. This path is determined by BGP for destinations outside the AS. This path would be chosen by the Label Distribution Protocol (LDP) if LDP was used between ASs.

3.2 Centralized Path Selection with CSPF

In this technique, the computation is performed by a single entity, that we name “global PCE”. We assume that the global PCE learns the complete topology by receiving the ISIS/OSPF link state packets of each AS. It performs a CSPF computation for each LSP. We note that such a computation does not rely on BGP. It is not constrained by BGP peering relationships and route filtering. This computation provides an indication of the path quality that can be achieved with a centralized computation.

Such a centralized solution could be envisaged when MPLS LSPs are entirely contained inside ASs that belong to the same company. However, it is not realistic for MPLS LSPs that cross ASs from different companies as this requires the ASs to cooperate and reveal their internal topology. Moreover, this solution is not scalable in the number of nodes and links of the ASs considered by the centralized computation. We use it as a benchmark and compare it with more easily deployable techniques.

3.3 ERO Expansion

Because the use of a global PCE performing CSPF computations is not applicable in the general interdomain framework, other techniques are required. In this section, we consider the use of RSVP-TE to establish interdomain MPLS LSPs.

Inside RSVP-TE, it is feasible to indicate the path or a portion of the path to be followed by the LSP inside an object called the Explicit Route Object (ERO). The ERO expansion technique, described in [12], relies on this object. It consists in completing at the ingress router of a domain, the ingress AS Border Router, the path computation up to the last reachable hop within the downstream domain, i.e. the BGP Next-Hop (NH). The computed path segment is then stored inside the ERO of the RSVP-TE Path message. This message is forwarded along the path specified inside the ERO and requests the establishment of the LSP along the path.

In addition to RSVP-TE signalling, we assume that there is a Path Computation Element (PCE) [5] inside each domain. The PCE is responsible for the computation of the paths on behalf of the ingress routers. It receives all the BGP routes learned inside the AS in order to improve the diversity of the routes available for the path computation [8].

Upon reception of an RSVP Path message requesting the establishment of an LSP, an AS Border Router (ASBR) sends a Path Computation Request (PCReq) to its PCE. After the completion of the computation, the PCE replies with a Path Computation Reply (PCRep) message. This message contains a path segment from the ingress ASBR to a BGP Next-Hop (NH) or indicates that there is no path segment respecting the constraints.

The ASBRs store the list of NHs that have already been tried for an LSP and lead to an infeasible path with regard to the constraints. When the PCE is not able to complete the path with a segment respecting the constraints, "crankback" is performed [4]. That is, the ASBR generates an RSVP Path Error message and sends it upstream. The upstream ASBR requests from its PCE the computation of a new segment avoiding the NHs that have already been tried.

The role of crankback is crucial for the establishment of interdomain LSPs because only limited information is available concerning the paths to reach a destination outside an AS. Thus, a PCE that computes a portion of a constrained interdomain LSP must rely on heuristics to choose an appropriate BGP NH among the NHs announced for the destination. If a bad choice is performed by the heuristic at some PCE, a downstream PCE may not be able to complete the computation of the path. Crankback enables to cope with such a situation and subsequently try alternative NHs.

In this paper, we propose two heuristics for the selection of the NHs by the PCEs during the computation of LSPs. The heuristics try to determine the NHs that are along short delay paths because the LSPs considered are subject to maximum end-to-end delay constraints in addition to bandwidth reservations.

Nearest NH. We call our first NH selection heuristic "nearest NH". Two link metrics are provided with ISIS-TE/OSPF-TE : the classical IGP metric and a TE metric. The IGP metric is usually set to the link bandwidth. We propose to set the TE metric of a link to its delay. Among the NHs available for the destination, the PCE selects the NH with the shortest path, from the ASBR to the NH, with enough bandwidth to support the LSP. The TE metric is used for the computation of the shortest path.

Vivaldi $2d + h$ Coordinates. Selecting the "nearest NH" in terms of the delay, as in the first heuristic, does not ensure that the end-to-end delay of the path will be low. The path segment downstream of a NH selected with the "nearest NH" heuristic may have a long delay. Thus, the heuristic proposed in this section relies on a delay estimation of the paths through the candidate NHs up to the tail-end of the LSP.

We use a virtual coordinate system, called Vivaldi [2], to estimate the delay of a path between two nodes. In this coordinate system each node computes its coordinates based on RTT measurements with a limited number of other nodes. Nodes connected with a low delay path will have neighboring coordinates while nodes connected through a higher delay path will be further apart.

In the heuristic presented in this section, we prefer to explore NHs that are along the path with the smallest delay estimation toward the tail-end D, to minimize the delay of the remaining portion of the path to D. Thus, for an ingress ASBR I_c inside an AS

AS_c, we prefer the ingress ASBR I_d inside a downstream AS AS_d such that

$$delay(I_c, I_d) + distance(I_d, D) = \min_{I_j \in NH}(delay(I_c, I_j) + distance(I_j, D))$$

where NH is the set of potential NHs for tail-end D, $delay()$ is the delay of the ISIS/OSPF path computed with the TE metric and $distance()$ is the distance between two points in the virtual coordinate space.

In our simulations, each node computes its coordinates in a two-dimensional Euclidean space augmented with an height, noted $2d + h$, as proposed in [2]. The distance between two nodes with coordinates (x_1, y_1, h_1) and (x_2, y_2, h_2) in the $2d + h$ space is the sum of the distance of the first node to the plane (its height, h_1), the Euclidean distance between the coordinates of the two nodes in the plane ($\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$) and the distance from the plane to the second node (the height of the second node, h_2).

In order to compute the preference of the candidate NHs, the PCE needs to know the coordinates of each NH and of the LSP's tail-end. For this purpose, we assume that after the computation of its coordinates, each node stores these coordinates inside its Domain Name Server (DNS), as proposed in [3]. The PCE requests the coordinates of the candidate NHs and the destination from the DNS.

In figure 1, we illustrate the selection of the NH by the two heuristics for an LSP entering $AS2$ at router $R2$ with tail-end $R8$. There are two candidate NHs, $R5$ and $R6$, for destination $R8$. The PCE inside $AS2$ prefers $R5$ over $R6$ with the "nearest NH" heuristic because the shortest delay path from $R2$ to $R5$ is 2 and the shortest delay path from $R2$ to $R6$ is 7. With the "vivaldi" heuristic, the PCE prefers $R6$ instead of $R5$ because the delay estimation[2] of the path from $R2$ to $R8$ transiting through $R6$ is $7 + \sqrt{(37 - 34)^2 + (18 - 10)^2} = 15.5$ and the delay estimation of the path transiting through $R5$ is $2 + \sqrt{(61 - 34)^2 + (78 - 10)^2} = 75$. The path from $R1$ to $R8$ obtained with the "nearest NH" heuristic is $R1 - R2 - R4 - R5 - R7 - R6 - R8$ with delay of 44 ms. On the other hand, the path $R1 - R2 - R4 - R3 - R6 - R8$, resulting from the computation with the "vivaldi" heuristic has a shorter delay of 9 ms.

4 Simulations

In this section, we present the results of simulations on two types of topologies[3]. First, we use topologies composed of 5 transit ASs to evaluate our heuristics in a small environment with MPLS deployed between the ASs. Such an environment is conceivable today. Then, we apply the path computation techniques on a larger topology composed of 20 transit ASs, as in the core of the Internet [11], to evaluate the techniques in a large scale deployment of inter-AS MPLS LSPs. We compare the four path selection techniques of section 3 in our simulations.

[2] In this example, we consider $2d$ coordinates. The delay estimation between two nodes in this $2d$ space is the Euclidean distance between the coordinates of the two nodes.

[3] The topologies and scripts used to provide the results presented in this section are available to the research community at the following URL: `http://totem.info.ucl.ac.be/tools`.

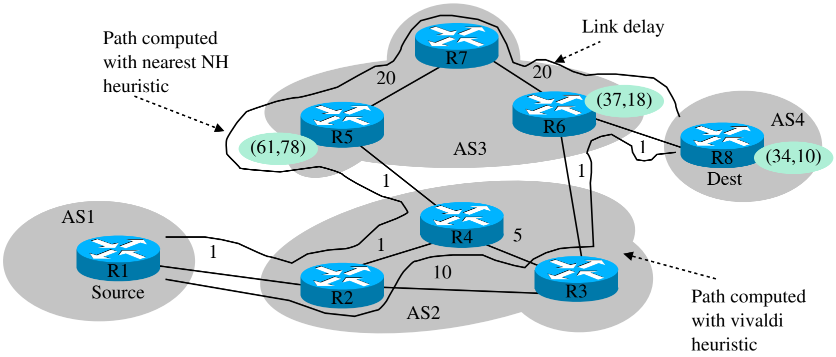


Fig. 1. Nearest NH versus vivaldi heuristics

4.1 Topologies

The topologies used for the simulations are generated with the transit-stub model of the GT-ITM tool [14]. First we generated 5 topologies each composed of 5 transit ASs. In these topologies, each transit AS is composed of approximatively 50 routers. The links inside the transit ASs are generated randomly with the parameters suggested by the authors of GT-ITM in [14]. GT-ITM attaches one stub AS to each router in a transit AS and randomly adds 250 extra links between the transit and the stub nodes. Each stub AS only contains one router. This router is the end-point of the LSPs established on the topology.

We group the stubs in classes that contain all the stubs attached to the same providers. We only keep one stub from each class to reduce the simulation time. It results in topologies with an average of 27 stubs. The nodes in these selected stubs and the nodes inside the transit ASs are placed by GT-ITM in an Euclidean plane. This placement is used to set the delay of the links. In our topology, the delay of a link is directly proportional to the Euclidean distance between its two end-points. In addition, we assign the same bandwidth to all the links.

In our simulations, we establish a full-mesh of LSPs between the routers in the stub ASs. Such a full-mesh could correspond to a very large interdomain BGP/MPLS VPN service. We establish the LSPs in one direction only. All LSPs are subject to the same bandwidth reservation (100 Mbps) and delay constraint (1900 ms). With a bandwidth reservation of 100 Mbps we can emulate the Fast-Ethernet service between Service Providers.

The delay constraint is determined as follows. For each LSP to be established, we computed the shortest path in terms of delay from the head-end to the tail-end node, on the complete topology and without BGP policies and filtering. We set the delay constraint of the LSPs to a round value just above the maximum delay of the resulting paths to ensure that, for each LSP, a path respecting the delay constraint exists in the topology.

We use the C-BGP simulator [9] to compute the BGP routing tables of the nodes. The routers inside stub ASs are configured not to advertise routes received from other ASs. Thus, stub ASs do not provide transit service. Transit ASs do not filter out the routes advertised to neighboring ASs. This ensures that each AS receives at least one route for each destination.

The second topology is composed of 20 transit ASs as the core of the Internet. It is generated by the method described earlier for the topologies with 5 transit ASs. Again, the transit ASs are composed of 50 nodes and all links have the same capacity. This topology has 411 stub ASs. We try to establish 84255 LSPs on this topology. Again all LSPs are subject to the same bandwidth reservation (100 Mbps) and end-to-end delay constraint (3300 ms).

4.2 Evaluation of the Path Selection Techniques

In this section, we present the results of the simulations on the topologies introduced in section 4.1. We first describe the results obtained from the simulations with the topologies containing 5 transit ASs. Then, we analyze the results obtained on the larger topology. In this analysis, we focus our attention on three aspects: the end-to-end delay of the LSPs, the number of LSPs that can be supported by the network, in our case this is proportional to the total amount of traffic that can be carried on the topology, and, finally, the amount of crankback that occurs during the computation of the constrained paths.

For each topology, we performed several simulations. The link bandwidths are set to a different value in each simulation. The objective is to study the impact of various levels of congestion on the LSP's establishment techniques. In the first simulation, the bandwidth of all links is set to 10 Gbps. Then, it is set to 2400 Mbps in the second simulation. Finally, it equals 622 Mbps in the third simulation.

In the remaining of this section, we distinguish the LSPs for which a path respecting the constraints could be found in the topology, called "established LSPs", from LSPs for which no suitable path could be found, called "failed LSPs".

The curves in figure 2 are obtained from simulations on the small topologies with link bandwidths set to 2400 Mbps and 622 Mbps. The results from the simulations with 10 Gbps links are similar to the results obtained with link bandwidths equal to 2400 Mbps. This is because there is no congestion in the topology with 10 Gbps links and only a few links are congested with 2400 Mbps links. In the latter topology, only 2 links are congested with the ideal CSPF computation inside the global PCE. Moreover, 10 links are congested with the "nearest NH" heuristic and, 0 links with "vivaldi".

Figures 2 (a) and 2 (b) show the cumulative distributions of the end-to-end delay for the different path computation techniques. They show, for a given delay on the x-axis, the number of established LSPs, on the y-axis, with end-to-end delay lower or equal to the value on the x-axis. Figures 2 (a) and 2 (b) present the results for a single topology, topology 0. Only the LSPs established on the topology are considered in these figures. The simulations performed with the other topologies with 5 transit ASs provide similar results.

In figure 2 (a), we first observe that there are more paths with a low delay with the CSPF computation performed by the global PCE than with the other techniques. We note that this computation does not rely on BGP. It is not constrained by BGP policies and filtering. It is used as an upper performance bound to which the other methods are compared. Moreover, there are more IP forwarding paths with a low delay than with the "nearest NH" and "vivaldi" heuristics. The good quality of the IP forwarding paths

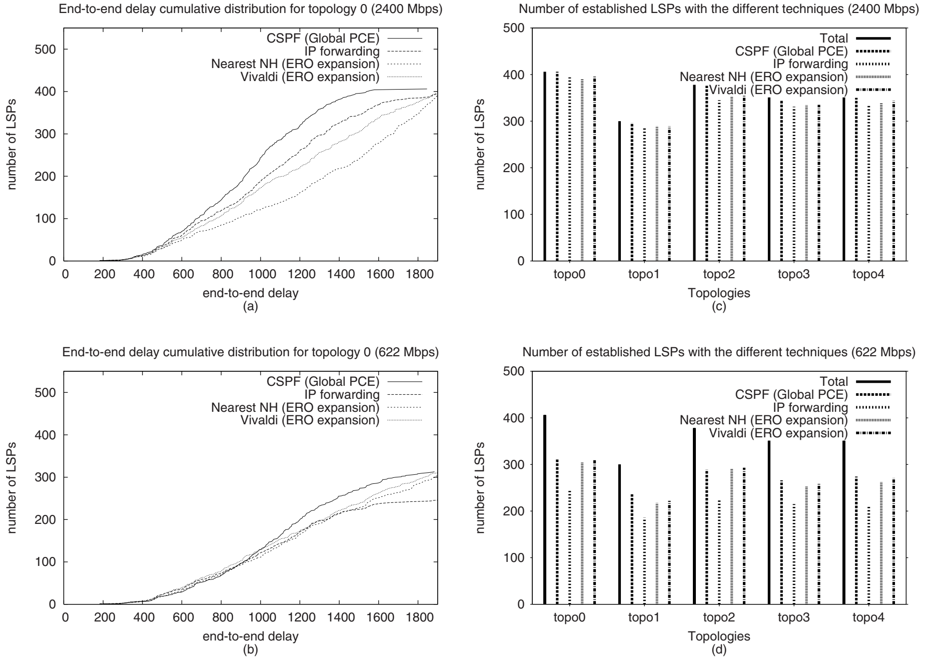


Fig. 2. Delay of LSPs established on topology with 5 transit ASs

in terms of delay comes from the fact that many BGP routes in our simulation are selected based on the IGP cost. Since we set the IGP cost of a link to its delay, the BGP selection rule based on the IGP cost prefers a route with a low delay over a route with a longer delay. Finally, the "vivaldi" heuristic provides more paths with a low delay than the "nearest NH" heuristic. This is due to the fact that "nearest NH" selects the NH only based on delay information that is local to the domain whereas the "vivaldi" NH selection is based on an estimation of the delay of the path that transits through the candidate NH.

When the bandwidth of the links is set to 622 Mbps in topology 0, congestion occurs on 4% of the links with CSPF and on 3% of the links with "nearest NH" and "vivaldi" path computation techniques. We observe in figure 2 (b) that there is not much difference between the 4 curves for the LSPs with end-to-end delay below 1000 ms. Above this value, there are more CSPF paths with a low delay compared to the other path computation techniques. Finally, we note that the total number of LSPs established along the IP forwarding paths is below the number of LSPs established with CSPF and our two heuristics. With IP forwarding, a router can only use a few outgoing interfaces for a destination. When the corresponding links are congested, the router is not able to send the path establishment request on an alternate link. Thus, the LSP establishment fails. However, the "nearest NH" and "vivaldi" heuristics rely on RSVP-TE for the establishment of the LSPs. RSVP-TE enables to avoid congested links in the establishment of an LSP by specifying the path to be followed by the LSP inside the Explicit Route Object (ERO) in order to bypass IP forwarding. Thus, techniques based on ERO expansion are more robust to congestion than standard IP forwarding. At last, we see that there are slightly more paths with a low delay with the “vivaldi” heuristic than with “nearest NH”. However this difference is not significant.

Figures 2 (c) and 2 (d) show the number of LSPs that can be successfully established on each topology, with the different path computation techniques. In figures (c) and (d), we observe that the number of LSPs established by the techniques relying on BGP routes, that is IP forwarding, “nearest NH” and “vivaldi”, is lower than with CSPF. The CSPF paths are computed by a centralized entity that possesses the complete topology. With BGP, however, only a portion of the routes available for a destination is distributed. Since there are fewer routes, they become faster congested. Moreover, some of these routes are selected by BGP based on other criterion than the delay. Thus, the resulting paths learned for a destination do not necessarily have a lower delay than the maximum end-to-end delay constraint of the LSPs.

We note that the number of LSPs established with the “vivaldi” heuristic is slightly higher than this number for the simulations with “nearest NH”. In both cases, the set of potential NHs depends on the BGP routes received for the destination. The set of potential NHs, inside an AS, is the same for many destinations. Among this set, the selection of the NH for a given ingress ASBR only relies on the delay of the shortest delay path with enough bandwidth for the LSP, in the “nearest NH” heuristic. Thus, the LSPs entering an AS through an ingress point incur the same delay until the shortest delay path becomes congested and a longer delay path is followed in the AS. Therefore, the delay incurred inside an AS by the LSPs entering at the same ingress ASBR increases as the LSPs are established. On the other hand, the selection of the NH by the “vivaldi” heuristic relies on the shortest delay path inside the AS and on the delay estimation from the NH to the destination of the LSP. Consequently, the delay incurred by the LSPs crossing an AS does not increase as fast as with the “nearest NH” heuristic because the LSP establishment requests entering an AS at an ingress point are distributed among multiple paths inside the AS based on the destination of the LSP. The delay of the paths computed with the “nearest NH” heuristic increases faster than with “vivaldi”. As a consequence, if all LSPs are subject to the same delay constraint, this constraint will be harder to fulfill with “nearest NH” than with “vivaldi”, as the LSPs are established.

Figure 2 (d) shows that in a less provisioned network, compared to the results in figure 2 (c), the number of LSPs that can be established along IP forwarding paths drops, as mentioned earlier. In addition, the number of LSPs established with our two ERO expansion heuristics is not far below the number of LSPs established along the CSPF paths computed by the global PCE. This is mostly due to the use of the RSVP-TE ERO expansion technique and the crankback mechanism.

The crankback mechanism enables to inform an upstream node of the failure during the establishment of an LSP and to try the establishment of the LSP along another path. Crankback occurs at most 12 times with “vivaldi” and 25 times with “nearest NH” for established LSPs, on topology 0 with link bandwidths set to 2400 Mbps. However, there are 95% of LSPs established with “vivaldi” without the help of crankback and 73% with “nearest NH”. On topology 0 with 622 Mbps links, there are 85% and 67% of the LSPs that are established without performing crankback with the “vivaldi” and “nearest NH” heuristics, respectively. Moreover, the maximum number of crankback for established LSPs is 13 for "vivaldi" and 14 for "nearest NH". We observe that, for the congested topology, crankback enables to carry more traffic inside the congested topology than when the IP forwarding paths are used. The contribution of the crankback mechanism, in the interdomain framework where the complete topology and the traffic load is not known by a single entity, is significant.

Now, we analyze the results of the simulations performed with the topology containing 20 transit ASs and 10 Gbps links. In these simulations, there are 13% of congested links with CSPF, and only 2% with both "vivaldi" and "nearest NH" heuristics. We observe from figure 3 (b) that the amount of traffic carried inside the topology is higher with CSPF than with our heuristics. There are 19% (18%), from the total amount of LSPs, of additional LSPs established with CSPF compared to the use of "nearest NH" ("vivaldi", respectively). Moreover, there is a difference of 32%, from the total number of LSPs, of established LSPs between CSPF and IP forwarding.

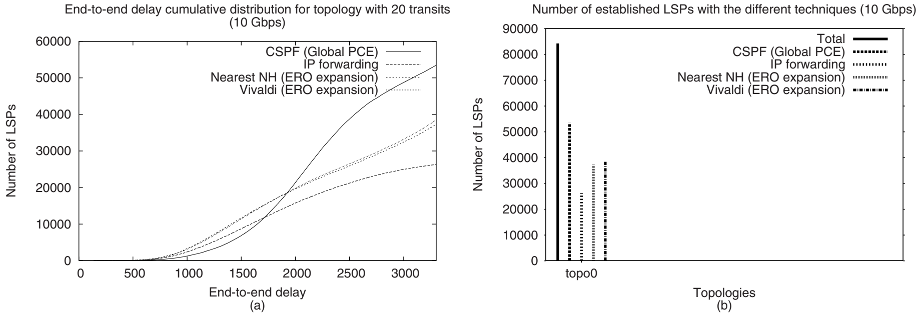


Fig. 3. Delay of LSPs established on topology with 20 transit ASs

Figure 3 (a) shows the cumulative distribution of the end-to-end delay for the LSPs established on the topology with the different path selection techniques. This distribution is almost the same for the two heuristics coupled with ERO expansion. However, we see that there are more paths with an end-to-end delay below 1927 ms, with our two heuristics than with CSPF. We assume that this is due to the higher number of LSPs established with CSPF than with the two heuristics. Some LSPs with a delay shorter than 1927 ms are established at the end of the simulation with our heuristics. However, with CSPF the low delay links are already congested. A short delay path may be found by the heuristics because there is less congestion in the topology than with CSPF due to the lower number of LSPs already supported by the topology.

On this large topology, crankback plays an important role. There are 45% of the established LSPs for which crankback occurs with "vivaldi" and 54% with "nearest NH". The maximum number of crankback for the establishment of an LSP is 199 with "vivaldi" and 283 with "nearest NH". However, there is less than 6 crankbacks for 90% of the LSPs established with "vivaldi" and less than 8 crankbacks, respectively, with "nearest NH".

5 Conclusion and Further Work

In this paper, we studied the establishment of constrained interdomain MPLS LSPs. We presented and evaluated four path computation techniques. Two of these techniques rely on heuristics proposed in this paper. The first technique tries to establish constrained MPLS LSPs along the default BGP route. The other techniques take advantage of RSVP-TE and the Path Computation Elements (PCEs) that are currently discussed within the IETF. First, we assume the existence of a global PCE that performs a CSPF computation on the complete topology for each LSP. This technique is not applicable in a general interdomain framework. It gives an upper performance bound for the other techniques. In the last two techniques, the computation of the constrained paths is distributed. Each PCE selects a NH to leave its AS based on the heuristics proposed in this paper and computes the path toward this NH.

Our simulations showed that using the default BGP route to establish constrained MPLS LSPs is not a good solution. A large amount of LSPs cannot be established. In addition, the simulations indicate that the number of constrained interdomain MPLS LSPs successfully established significantly increases with the two heuristics. Moreover, the "vivaldi" heuristic is slightly better than "nearest NH". More LSPs are established with "vivaldi" and the maximum amount of crankback is lower. However, this negligible improvement has a cost. It requires the computation of coordinates. Finally, we saw that the amount of crankback during the establishment of the LSPs with both heuristics is low for a very large portion of the LSPs. This is a strong argument in favor of ERO expansion for the current standardization work within the IETF.

In this paper, we presented and evaluated two heuristics for the "ERO expansion" architecture described in [5]. [5] also proposes another architecture that relies on communication between PCEs in order to find a constrained path for an LSP. If the list of ASs to be crossed by the LSP is not known a priori, the heuristics of this paper may be used to select a subset of the downstream ASs and, thus, of the PCEs that will contribute to the path computation. We propose to evaluate such a solution in the future.

Acknowledgements

The authors thank Steve Uhlig, Virginie Van den Schrieck and Pierre Francois for their reviews as well as Bruno Quoitin for providing the C-BGP tool. The authors thank Cédric de Launois for the code related to the computation of the vivaldi coordinates.

References

1. M. Boucadair. QoS-Enhanced Border Gateway Protocol. Internet draft, draft-boucadair-qos-bgp-spec-00.txt, work in progress, June 2005.
2. F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of the ACM SIGCOMM '04 Conference*, Portland, Oregon, August 2004.
3. C. de Launois. *Unleashing Traffic Engineering for IPv6 Multihomed Sites*. PhD thesis, Université catholique de Louvain, September 2005.

4. A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash. Crankback signaling extensions for MPLS and GMPLS RSVP-TE. Internet draft, draft-ietf-ccamp-crankback-05.txt, work in progress, May 2005.
5. A. Farrel, J-P. Vasseur, and G. Ash. Path computation element (PCE) architecture. Internet draft, draft-ietf-pce-architecture-02.txt, work in progress, September 2005.
6. N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In *ACM SIGCOMM workshop on Future Directions in Network Architecture (FDNA 2004)*, August 2004.
7. Bradley Huffaker, Marina Fomenkov, Daniel J. Plummer, David Moore, and k claffy. Distance metrics in the internet. In *IEEE International Telecommunications Symposium*, 2002.
8. C. Pelsser, S. Uhlig, and O. Bonaventure. On the difficulty of establishing interdomain LSPs. In *IEEE International Workshop on IP Operations and Management (IPOM 2004)*, Beijing, China, October 11-13th 2004.
9. B. Quoitin and S. Uhlig. Modeling the Routing of an Autonomous System with C-BGP. *IEEE Network*, 19(6), November 2005.
10. N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking*, 12(1):2–16, February 2004.
11. L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *INFOCOM 2002*, June 2002.
12. J-P. Vasseur, A. Ayyangar, and R. Zhang. A per-domain path computation method for computing inter-domain traffic engineering (TE) label switched path (LSP). Internet draft, draft-ietf-ccamp-inter-domain-pd-path-comp-00.txt, work in progress, April 2005.
13. M. Yannuzzi, S. Sánchez-López, X. Masip-Bruin, J. Solé-Pareta, and J. Domingo-Pascual. A combined intra-domain and inter-domain qos routing model for optical networks. In *9th conference on Optical Network Design and Modelling (ONDM 2005)*, Milan, Italy, February 7-9th 2005.
14. Ellen W. Zegura, Kenneth L. Calvert, and Michael J. Donahoo. A quantitative comparison of graph-based models for Internet topology. *IEEE/ACM Transactions on Networking*, 5(6):770–783, 1997.

Reliable Routings in Networks with Generalized Link Failure Events[⋆]

Stamatis Stefanakos

Dept. of Computer Science,
University of Rome "La Sapienza," 00198 Rome, Italy
stefanak@di.uniroma1.it

Abstract. We study routing problems in networks that require guaranteed reliability against multiple correlated link failures. We consider two different routing objectives: The first ensures "local reliability," i.e., the goal is to route so that each connection in the network is as reliable as possible. The second ensures "global reliability," i.e., the goal is to route so that as few as possible connections are affected by any possible failure. We exhibit a trade-off between the two objectives and resolve their complexity and approximability for several classes of networks. Furthermore, we propose approximation algorithms and heuristics. We perform experiments to evaluate the heuristics against optimal solutions that are obtained using an integer linear programming solver. We also investigate up to what degree the routing trade-offs occur in randomly generated instances.

1 Introduction

As high-speed networks become widely deployed and more commercial services depend on them, it is extremely important to provide reliable connections to the end users. In circuit-switched networks, connections are established by reserving resources along end-to-end paths that are kept up for the duration of the communication. Such networks are, for example, the so-called all-optical networks [12]. In all-optical networks, connections are established through light-paths and several light-paths are multiplexed in the same optical fiber that can carry data at rates of the magnitude of Terabits/sec. In these networks, reliable communications are even more crucial since a short network outage can lead to massive data losses and can affect many connections.

Traditional methods for ensuring reliable transmissions in circuit-switched networks rely on the precomputation of a backup path for every working path or for every network link (see [9] and [15] and the references therein for studies of single link protection in all-optical networks). These methods work fine as long as the network experiences only single link failures. They do not guarantee undisturbed communication, however, in the case of multiple link failures. Such failures are not seldom and often are correlated: a single failure in the physical

[⋆] Work partially done while the author was with the Computer Engineering and Networks Laboratory of the Swiss Federal Institute of Technology in Zurich.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 221–232, 2006.

network (a cut in the conduit carrying wiring or fibers used for several links) results in several failures in the abstract network layer (see [6] for a discussion on multiple link failures). This type of link failures can be modeled using the notion of *generalized failure events.* A single generalized failure leads to the failure of several links in the network. Links that belong to the same failure event are also said to be in the same shared risk link group [5].

In this paper, we consider the problem of computing reliable routings in networks with generalized failure events under two different notions of reliability. In what we call "local reliability," we seek routings in which each connection spans as few as possible different failure events. In a sense, we minimize the failure probability of each connection, assuming that all failure events occur equally likely. In what we call "global reliability," we seek routings in which any single failure event affects as few as possible connections. Under this objective, we are interested in minimizing the distortion to the network operation in case of a failure event.

1.1 Problem Definitions and Preliminaries

We model the network via an undirected multi-graph $G = (V, E)$. Connection requests are pairs of vertices in the graph. A connection (s, t) is established via an undirected simple path from s to t (an s-t path). We represent failure events by assigning colors (labels) to the edges of the graph. That is, each failure event corresponds to a single color and all edges baring a particular color fail when the corresponding event occurs.

We consider "locally" reliable routings, in the sense that each single established connection should be as reliable as possible. Formally, we speak about the Minimum Color Path problem: We are given an undirected multi-graph $G = (V, E)$, an assignment of colors to the edges of G, and a set of connection requests $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$. We seek a *routing* of R, i.e., an $s_i - t_i$-path for $1 \le i \le k$. The goal is to minimize the average number of colors contained in a path. Observe that the problem can be solved for each path separately: we simply seek a path with the minimum number of colors for each request.

"Globally" reliable routings are routings in which any failure event only affects a few of the connections. Formally, we speak about the Minimum Color Congestion Routing problem: We are given an undirected multi-graph $G = (V, E)$, an assignment of colors to the edges of G, and a set of connection requests $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$. We seek a routing of R, such that the maximum number of paths that contain the same color is minimized.

The Minimum Color Congestion Routing problem is closely related to the Minimum Load Routing problem. The latter asks for a routing so that the maximum number of paths per edge (i.e., the maximum load) is minimized. Minimum Color Congestion Routing is a generalization of Minimum Load Routing and it reduces to the special case if each edge is assigned a different color.

We give some formal definitions regarding algorithms and graph classes that we will need later on: An algorithm A for a minimization problem Π is a ρ-*approximation algorithm* if for every instance I of Π, A runs in time polynomial in $|I|$ and delivers a solution with objective value at most $\rho \cdot OPT(I)$, where $OPT(I)$ denotes the objective value of an optimal solution for I. We also say that ρ is the *approximation ratio* of algorithm A. A graph is a *chain* if it is a path (a multi-chain can have multiple edges between adjacent vertices; in this paper when we refer to a chain it is implied to be a multi-chain). A graph is a *ring* if it is a cycle. A (multi-)graph is a *series-parallel graph* if it can be obtained from a collection of trees by repeatedly replacing edges by parallel edges or by chains. We refer the reader to [11] for formal definitions of complexity classes.

1.2 Related Work

The topics of path protection and network survivability have received significant attention in the last years, especially for all-optical networks. We do not provide a comprehensive survey of the literature here but rather focus on work that closely relates to the optimization problems that we study. Note that as, to the best of our knowledge, the Minimum Color Congestion Routing problem has not been studied before, we will review only results related to the Minimum Color Path problem.

Carr et al. [3], motivated by a data mining application, introduce the Red-Blue Set Cover problem: given a set $R = \{r_1, \ldots, r_{|R|}\}$ of red elements, a set $B = \{b_1, \ldots, b_{|B|}\}$ of blue elements, and a family $\mathcal{S}$ of subsets of $R \cup B$, one seeks a subfamily of $\mathcal{S}$ that covers all blue elements but covers only the minimum possible number of red elements. Carr et al. show that this problem reduces to Minimum Color Path. They also show that the Red-Blue Set Cover can not be approximated within a ratio of $O(2^{\log^{1-\varepsilon} n})$ for any positive ε, unless $NP \subseteq DTIME(n^{\text{polylog}(n)})$, where $n = |\mathcal{S}|^4$. This translates to a similar inapproximability result for Minimum Color Path: unless $NP \subseteq DTIME(n^{\text{polylog}(n)})$, no $O(2^{(\log k)^{1-\varepsilon}})$-approximation algorithm for any positive ε can exist for Minimum Color Path, where $k \in \Omega(n^{1/4})$ [14]. Note that the graph in the constructed instance of Minimum Color Path in the reduction of Red-Blue Set Cover to Minimum Color Path given in [3] is series-parallel; thus, this hardness result applies already to instances restricted to series-parallel graphs.

Yuan, Varma, and Jue [16] show that the Minimum Color Path problem is NP-hard in chains. They propose heuristics for general graphs which they evaluate using experiments on randomly generated instances. They also study variations of Minimum Color Path in which for each request two disjoint paths are sought that have the minimum number of total colors, or the minimum color overlap.

Other authors have studied the problem of establishing a spanning tree using as few colors as possible. Chang and Leu [4] prove this problem to be NP-hard and propose two heuristics and an exact exponential-time algorithm. Krumke and Wirth [8] analyze the performance of the two heuristics of [4]. They show that one of the two can be arbitrarily bad while the other achieves a logarithmic approximation ratio. Wan, Chen, and Hu [13] slightly improve the approximation ratio shown by Krumke and Wirth [8].

1.3 Our Results

We begin, in Section 2, by exhibiting trade-offs between the different routing objectives. We show that routing with respect to local reliability can be arbitrarily bad for global reliability and vice versa. Also, we consider failure-oblivious routings and show that they can be arbitrarily bad with respect to the reliability objectives.

In Section 3, we consider the Minimum Color Path problem. We give approximation algorithms achieving logarithmic approximation ratio for the special cases of chains, rings, and trees. We also show that this is best possible by providing a matching inapproximability result.

In Section 4, we consider the Minimum Color Congestion Routing problem. We obtain inapproximability results for general network topologies using the existing results for the Minimum Load Routing problem. Furthermore, we show that Minimum Color Congestion Routing is NP-hard even in the special case of chain networks and can not be approximated within a factor better than 2 unless $P = NP$. Finally, we present a heuristic for Minimum Color Congestion Routing that routes the requests sequentially along shortest paths that are computed with respect to edge-weights that are exponential in the current maximum color congestion.

We present experimental results in Section 5. We use integer linear programming formulations of Minimum Color Path and Minimum Color Congestion Routing to obtain optimal solutions for the two problems.

We note that several technical details and proofs have been omitted from this version due to space limitations.

2 Routing Trade-Offs

2.1 Local vs. Global Reliability

Observe that while the Minimum Color Congestion Routing problem tries to minimize the *maximum* number of paths per color, the Minimum Color Path problem seeks to minimize the *average* number of paths per color. This renders the two problems considerably different: it is easy to construct instances where the optimal solution for one is suboptimal for the other. We omit the details from this version because of space limitations.

2.2 Oblivious vs. Failure-Aware Routings

Oblivious routing is done without taking into consideration the link failure events of the network, i.e., routing is performed assuming that each edge fails independently of any other. This can be a choice of the network designer in order to simplify the network protocols or it can be because of lack of information. We are interested to see how “bad” such oblivious routings can be with respect to the best possible routings if we take into consideration the failure events of the network. It can be shown that routing under limited knowledge can be arbitrarily bad with respect to both routing objectives. Nevertheless, failure-aware routings can be arbitrarily bad with respect to traditional routing objectives. It is fairly easy to see that routing with the objective of Minimum Color Path can produce routings with very large link-load and very large delays (hop-count). Minimizing the color congestion can also result in similar routings. In Section 5, we will investigate whether such routings occur in practice.

3 Locally Reliable Routings

As we have discussed in Section 1.2, already in series-parallel graphs there can not exist any reasonable approximation algorithm for Minimum Color Path, unless $NP \subseteq DTIME(n^{\mathrm{polylog}(n)})$ (i.e., all problems in NP can be solved by deterministic algorithms that run in quasi-polynomial time) [14]. In the simplest topologies, while we can still not hope for constant approximation algorithms, we can achieve logarithmic approximation algorithms, as the following two theorems show.

Theorem 1. *There exists an $O(\log |V|)$-approximation algorithm for* Minimum Color Path *in chains, rings, and trees.*

Theorem 2. Minimum Color Path *can not be approximated within a ratio of $c \log |V|$ for some positive constant c, unless $P = NP$, even in chains.*

4 Globally Reliable Routings

It is easy to see that Minimum Color Congestion Routing is a generalization of the Minimum Load Routing problem. Given an instance of Minimum Load Routing we can reduce it to Minimum Color Congestion Routing simply by assigning a different color to every edge of the graph. In [1], Andrews and Zhang show that Minimum Load Routing can not be approximated within a ratio of $(\log \log m)^{1-\varepsilon}$ for any positive ε, unless $NP \subseteq ZTIME(n^{\mathrm{polylog}(n)})$, where m is the number of edges of the input graph. We thus obtain the following hardness result for Minimum Color Congestion Routing.

Theorem 3. *The problem* Minimum Color Congestion Routing *can not be approximated within a ratio of $(\log \log |E|)^{1-\varepsilon}$ for any positive ε, unless $NP \subseteq ZTIME(n^{\mathrm{polylog}(n)})$.*

Even in the very simple topology of chains, the problem remains NP-hard.

Theorem 4. Minimum Color Congestion Routing *is NP-hard in chains.*

The proof uses a reduction from the Domatic Number problem. (The *domatic number* of a graph $G = (V, E)$ is the maximum number k such that V can be partitioned into k dominating sets. A dominating set $V' \subseteq V$ for a graph $G = (V, E)$ has the property that every vertex $v \in V$ is in V' or has a neighbor in V'.) One can also show the following:

Theorem 5. Minimum Color Congestion Routing *can not be approximated within a ratio of $2 - \varepsilon$ for any positive ε in chains, unless $P = NP$.*

Observe that if each failure event affects only a constant number of links in the network, then we can employ an algorithm for Minimum Load Routing and lose only a constant factor in the approximation ratio:

Theorem 6. *A ρ-approximation algorithm for* Minimum Load Routing *implies a $c \cdot \rho$-approximation algorithm for* Minimum Color Congestion Routing *for the case where every color is used in at most c edges.*

Given the inherent intractability of the Minimum Color Congestion Routing problem, we propose a heuristic to tackle it in the general case. The heuristic is shown below (Algorithm 1). It computes a shortest path for each of the requests. At every iteration one request is routed. The path assigned to each request is the shortest path with respect to edge-weights which are exponential in the current congestion of each color. When a path is assigned to a request, it does not change until the end of the execution.

Algorithm 1. Heuristic for Minimum Color Congestion Routing

```
Input: Graph $G = (V, E)$, color assignment $\mu : E \to \{1, \dots, c\}$, set of requests $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$
Output: Set of paths $P = \{p_1, \dots, p_k\}$
  Guess optimal color congestion $L^*$
  $\Lambda := 2L^* \log(2m)$
  for $i = 1$ to $k$ do
    $L_j(i) :=$ Load of color $j$ before routing request $i$
    $w_e(i) := (2 \cdot |E|)^{L_{\mu(e)}(i)/\Lambda}$
    $p_i :=$ shortest $s_i$-$t_i$ path w.r.t. the edge-weights $w_e(i)$
  end for
```

The heuristic is an adaptation of the $O(\log |V|)$-approximation algorithm for Minimum Load Routing by Aspnes et al. [2]. The difference with the original algorithm is that for Minimum Color Congestion Routing we use weights that are exponential in the congestion of each color and not in the link load. Also, note the use of the parameter L^*, which should be equal to the optimal load. This can be guessed by trying all k possible values (since the optimal congestion lies between 1 and $|R| = k$) and then picking the best solution.

5 Experiments

5.1 Goal of the Experiments

The experiments have two goals: First, we want to investigate up to what degree the worst-case trade-off behavior of the instances described in Section 2 occurs in randomly generated instances. Second, we want to see how far from the optimal are the routings returned by the heuristic for Minimum Color Congestion Routing.

The concrete questions we want to answer are the following:

- How does the algorithm for Minimum Load Routing of Aspnes et al. [2] perform with respect to the Minimum Color Path and Minimum Color Congestion Routing objectives?
- How does a simple routing algorithm that routes every request along a shortest path perform in practice with respect to the Minimum Color Path, the Minimum Color Congestion Routing, and the Minimum Load Routing objectives?
- How does the heuristic algorithm for Minimum Color Congestion Routing compare to an optimal algorithm for Minimum Color Congestion Routing? Also, how does it perform with respect to the Minimum Load Routing and Minimum Color Path objectives? What is the average length of the routings returned by the heuristic and how do they compare to the shortest average length?
- How does an optimal algorithm for Minimum Color Path perform with respect to the Minimum Load Routing and Minimum Color Congestion Routing objectives? What is the average length of Minimum Color Path routings and how do they compare to the shortest average length?

To get optimal solutions for Minimum Color Congestion Routing and Minimum Color Path we use a general purpose integer linear program solver. The formulations are straight-forward and are omitted from this version.

5.2 Implementation Details

The implementation was done in C++ using LEDA [10] (version 4.3.1) for the basic data types and graph algorithms and CPLEX [7] (version 8.1) for solving the integer linear programs. The code was compiled with the GNU C++ compiler (version 2.95.3) on SunOS 5.8. All experiments were run on a SunFire 480R with 4GB RAM and two 900 MHz processors (our code uses only one processor) each with 8 MB (4ns) L2 Cache. The workstation was available to other users as well during the experiments.

5.3 Experimental Results

For the remainder of this section, we will use the following notation to refer to the tested algorithms: MLR stands for the $O(\log |V|)$-approximation algorithm for Minimum Load Routing by Aspnes et al. [2]; SPR stands for the simple algorithm that routes each request through some shortest path; MCCR stands for our heuristic (Algorithm 1) for Minimum Color Congestion Routing; MCCRIP stands for the ILP-based optimization algorithm for Minimum Color Congestion Routing; finally, MCPIP stands for the ILP-based optimization algorithm for Minimum Load Routing.

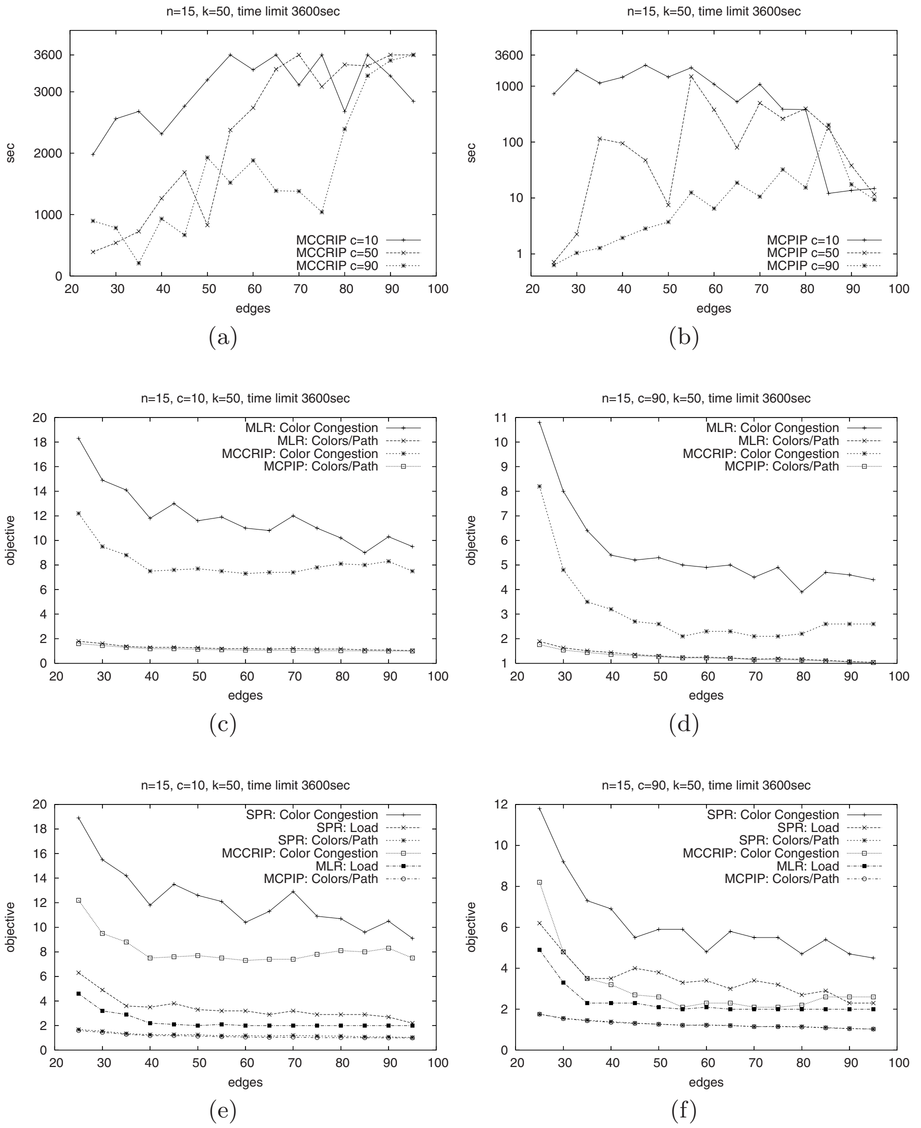
We begin with a brief discussion on how the experiments to be performed were chosen. An important limitation in the design of the experiments has been the running time of MCCRIP and MCPIP, i.e., the CPLEX optimization time.

We chose to set an upper bound of one hour to the optimization time in order to be able to perform a large number of experiments in a reasonable amount of time. If this amount of time elapses, CPLEX either returns a possibly sup-optimal integer solution if it has found one or exits without returning a solution. In a preliminary experiment on instances with 50 nodes, 500 edges, and 50 requests, MCCRIP exceeded the time limit in all 20 instances that we tested with more than ten colors without producing an integer solution. MCPIP was faster: in the same experiment it found an optimal solution in seven out of 20 runs and returned a sub-optimal integer solution when it exceeded the time limit in the remaining 13 runs. In order to be able to get optimal or good sub-optimal solutions from the ILP optimizations we restricted the experiments to relatively small-sized instances. We performed experiments on networks with 15–65 nodes, 25–95 edges, 10–90 colors, using 50 requests for each instance. Each experiment was performed ten times and the plots that follow show the average of these executions. With these instance sizes, and with a time limit of one hour, we were able to solve all instances, although some not optimally. We give plots of the running times of MCCRIP and MCPIP in Figures 1(a) and 1(b). Both algorithms require more computation time as the number of colors decreases. MCPIP runs generally faster than MCCRIP.

In Figures 1(c) and 1(d) we have plotted the maximum color congestion and the average number of colors per path in the solutions of MLR along with the corresponding solutions from MCCRIP and MCPIP. The network consists of 15 nodes with edges ranging from 15 to 95, 50 requests, and 10 (Figure 1(c)) and 90 colors (Figure 1(d)). One notices immediately that the number of colors per path in the routings of MLR is roughly the same with that of MCPIP. The actual number is pretty low, around and below two colors per path (and decreases as the graph becomes denser). This is due to the small size of the instances (15 nodes) and their small diameter. (The number of paths per color increases as we increase the network size.) With respect to the color congestion, the routings from MLR are by around a factor of 3/2 away from MCCRIP in sparse graphs, but improve as the graphs get denser. We can conclude that routing to minimize the load is a reasonable choice for producing routings with small number of colors per path. Minimum load routings, however, can have large color congestion, especially in sparse graphs.

In Figures 1(e) and 1(f), we have plotted the maximum color congestion, the maximum load, and the average number of colors per path in the solutions of SPR along with the corresponding values from MCCRIP, MLR, and MCPIP. The network consists of 15 nodes with edges ranging from 15 to 95, 50 requests, and 10 (Figure 1(e)) and 90 colors (Figure 1(f)). Algorithm SPR performs similarly to MLR regarding the color congestion and the number of colors per path. The maximum load of SPR is by around a factor of 3/2 away from MLR in sparse graphs, but gets closer to MLR as the graphs get denser.

In Figures 2(a) and 2(b), we have plotted the maximum color congestion, the maximum load, the average number of colors per path, and the average hop-length in the solutions of MCCR along with the corresponding objective values



(a) (b) (c) (d) (e) (f)

Fig. 1. (a), (b): Running times of MCCRIP and MCPIP. (c), (d): Performance of MLR with respect to the Minimum Color Congestion Routing and Minimum Color Path objectives. (e), (f): Performance of SPR with respect to the Minimum Color Congestion Routing, Minimum Color Path, and Minimum Load Routing objectives.

from MCCRIP, MLR, MCPIP, and SPR. The network consists of 15 nodes with edges ranging from 15 to 95, 50 requests, and 10 (Figure 2(a)) and 90 (Figure 2(b)) colors. The routings of MCCR have color congestion which is really close

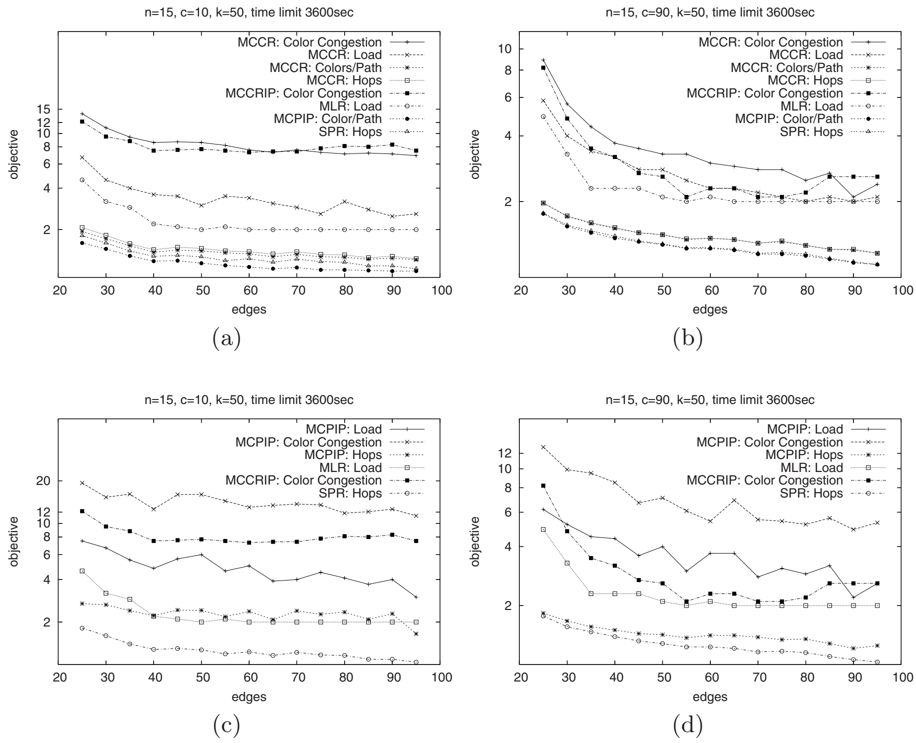


Fig. 2. (a), (b): Performance of MCCR with respect to the maximum color congestion, the maximum load, the average number of colors per path, and the average hop-length. (c), (d): Performance of MCPIP with respect to the maximum color congestion, the maximum load, and the average hop-length.

to that of MCCRIP. Actually, in some instances MCCR performs better than MCCRIP. This happens mostly in instances with ten colors which are the hardest to solve and MCCRIP more often than not exceeds the time limit of one hour and returns a suboptimal solution. Furthermore, observe that the load of MCCR is close to that of MLR (and gets closer in dense graphs). Also, the number of colors per path in the solutions of MCCR is very close to that of MCPIP, and the average hop-length is also close to that of SPR. Thus, the heuristic for Minimum Color Congestion Routing performs very well with respect to the maximum color congestion objective and produces reasonable routings with respect to the maximum load, the average number of colors per path, and the average hop-length objectives.

In Figures 2(c) and 2(d), we have plotted the maximum color congestion, the maximum load, and the average number of hops in the solutions of MCPIP along with the corresponding objective values from MCCRIP, MLR, and SPR. The network consists of 15 nodes with edges ranging from 15 to 95, 50 requests, and 10 (Figure 2(c)) or 90 (Figure 2(d)) colors. MCPIP produces routings with color congestion which is away from that of MCCRIP by a factor of around 3/2. The load of MCPIP is also higher than that of MLR and in sparse instances with many colors it even exceeds the color congestion of MCCRIP. It remains however within a factor of 2 from the load of MLR. As expected, MCPIP produces routings with average hop-length close to that of SPR in instances with 90 colors; in instances with 10 colors MCPIP produces routings with greater average hop-length than that of SPR (but within a factor of 2). Hence, MCPIP performs reasonably well with respect to the maximum load, the maximum color congestion, and the average hop-length objectives.

6 Conclusion

We have studied routing problems in networks with generalized failure events. We have analyzed the complexity and the approximability of a local (Minimum Color Path) and a global (Minimum Color Congestion Routing) routing objective for several classes of networks. We have also exhibited trade-offs between the two objectives as well as between the reliability objectives and traditional routing objectives such as minimizing the maximum congestion. Experiments on random instances have shown that the worst-case trade-offs do not occur in practice. Our experiments have also shown that the heuristic algorithm that we proposed to tackle Minimum Color Congestion Routing gives routings with maximum color congestion, maximum load, and average number of paths all very close to the optimal ones.

Several interesting directions for future work can be followed. From an algorithmic point of view it is very interesting to further study the Minimum Color Congestion Routing problem. Providing a logarithmic approximation algorithm for the general case, or a better inapproximability result would be a first step in that direction. Approximation algorithms for specific graph classes such as chains or rings would also be of great interest. From a practical point of view, it is interesting to study these objectives under additional routing objectives. For example, requiring disjointness among certain requests (e.g., for every request one might want to compute a primary path and a disjoint back-up path) is an important extension to consider.

References

1. M. Andrews and L. Zhang. Hardness of the undirected congestion minimization problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, 2005.
2. J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM*, 44(3):486 – 504, 1997.
3. R. D. Carr, S. Doddi, G. Konjevod, and M. Marathe. On the red-blue set cover problem. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '00)*, pages 345 – 353, 2000.

4. R.-S. Chang and S.-J. Leu. The minimum labeling spanning trees. *Information Processing Letters*, 63(5):277–282, 1997.
5. S. Chaudhuri, G. Hjálmtýsson, and J. Yates. Control of lightpaths in an optical network. IETF Internet Draft, March 2000.
6. H. Choi, S. Subramaniam, and H.-A. Choi. On double-link failure recovery in WDM optical networks. In *Proceedings IEEE INFOCOM 2002, The 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 808–816, 2002.
7. ILOG CPLEX. *CPLEX 8.1*, 2004. http://www.cplex.com/.
8. S. Krumke and H.-C. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66(2):81–85, 1998.
9. M. Médard, R. A. Barry, S. G. Finn, W. He, and S. Lumetta. Generalized loopback recovery in optical mesh networks. *IEEE/ACM Transactions on Networking*, 10(1):153–164, 2002.
10. K. Mehlhorn and S. Näher. *LEDA: A platform for combinatorial and geometric computing*. Cambridge University Press, 1999.
11. C. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.
12. R. Ramaswami and K. N. Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann Publishers, 2nd edition, 2002.
13. Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *Information Processing Letters*, 84:99–101, 2002.
14. H.-C. Wirth. *Multicriteria Approximation of Network Design and Network Upgrade Problems*. PhD thesis, University of Würzburg, 2001.
15. Y. Xin and G. N. Rouskas. A study of path protection in large-scale optical networks. *Photonic Network Communications*, 7(3):267–278, 2004.
16. S. Yuan, S. Varma, and J. P. Jue. Minimum-color path problems for reliability in mesh networks. In *Proceedings of IEEE INFOCOM 2005, The 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005.

Making Outbound Route Selection Robust to Egress Point Failure*

Mina Amin, Kin-Hon Ho, Michael Howarth, and George Pavlou

Centre for Communication Systems Research, University of Surrey, UK
{M.Amin, K.Ho, M.Howarth, G.Pavlou}@eim.surrey.ac.uk

Abstract. Offline inter-domain outbound Traffic Engineering (TE) can be formulated as an optimization problem whose objective is to determine primary egress points for traffic exiting a domain. However, when egress point failures happen, congestion may occur if secondary egress points are not carefully determined. In this paper, we formulate a bi-level outbound TE problem in order to make outbound route selection robust to egress point failures. We propose a tabu search heuristic to solve the problem and compare the performance to three alternative approaches. Simulation results demonstrate that the tabu search heuristic achieves the best performance in terms of our optimization objectives and also keeps traffic disruption to a minimum.

1 Introduction

Inter-domain Outbound Traffic Engineering (TE) [1,2] aims to control traffic exiting a domain by assigning the traffic to the best egress points (i.e. routers or links). Since inter-domain links are the most common bottlenecks in the Internet [2], optimizing their resource utilization is a key objective of outbound TE. In the literature, several outbound TE approaches have been proposed [2,3]. These proposals, however, have neglected the detrimental impact of inter-domain EP failure on the achieved TE performance. In fact, the network performance under failure conditions should ideally be optimized by considering failure as part of the outbound TE optimization.

Failure occurs as part of daily network operations [6]. Inter-domain failures are typically caused by: (1) *physical failures* such as inter-domain link fiber cut and equipment failure, or (2) *logical failures* such as router CPU overload, operation systems problem and maintenance. A recent study [4] discovered that logical inter-domain link failures are common events and are usually transient in nature. When a failure happens on an EP, traffic is shifted to another available EP in accordance to the BGP route selection policies. However, if a large amount of traffic is shifted, congestion is likely to occur on these new serving EPs. This problem has not been considered in the existing outbound TE proposals. An intuitive approach to minimize this congestion is to redirect the traffic to another EP by adjusting BGP routing policies in an online manner until the best available EP has been found. Such online trial-and-error approach may cause router misconfiguration, unpredicted traffic disruption

* This work was undertaken in the context of FP6 Information Society Technologies AGAVE (IST-027609) project, which is partially funded by the Commission of the European Union.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 233 – 246, 2006.

and BGP route flooding, leading to route instability. As a result, an outbound TE approach that produces optimal performance under both normal and failure scenarios so as to minimize online and unpredictable route changes is highly desirable.

In this paper, we propose a *multi-level* outbound TE approach that is robust to EP failure, which achieves reasonably good performance under both Normal State (NS) and Failure States (FS). We refer NS to no failure and each FS to a single EP failure. In multi-level outbound TE, the first level is to select Primary EPs (PEP) under NS, in a similar fashion to previous work [2,3]. Then, the second level is to select the next best EP as the Secondary EP (SEP) when the PEP fails. This approach can also be repeated for successive EPs. For example, a tertiary EP is used as the traffic exit point when both the PEP and SEP fail. However, since *single* link failure is the predominant form of failure in communication networks [6], we therefore consider a bi-level outbound TE formulation. This problem can be formulated as follows:

Given a network topology, destination prefixes and an inter-domain Traffic Matrix (TM), determine for each traffic demand the PEP and the SEP upon PEP failure. The optimization objective is to minimize the maximum EP utilization under Normal State (NS) and the average of maximum EP utilization across all Failure States (FSs).

Previous work [5,6,7] on making TE robust to link failure has focused on the intra-domain problem. Heuristics have been proposed to compute a set of IGP link weights that is robust to any single intra-domain link failure. Our work is similar to this previous work, but the primary difference is that we focus on inter-domain outbound TE. Given that a significant amount of Internet traffic is routed across domains (e.g. the rapidly increasing peer-to-peer traffic) and inter-domain link failures are common and transient, making outbound TE robust to failures is an important problem. To the best of our knowledge, this issue has yet not been investigated.

To solve the bi-level outbound TE problem, we propose a tabu search heuristic and compare its performance to alternative strategies. Experimental results demonstrate that the tabu search heuristic significantly improves the performance under all FSs (about 1%-3% from the FS lower bound) with a small performance degradation under NS (about 2%-8% from the NS lower bound). The tabu search heuristic also minimizes traffic disruption.

This paper has the following structure. Section 2 presents the bi-level outbound TE problem formulation. We detail the proposed tabu search heuristic in Section 3. Section 4 presents three alternative strategies for solving the problem. Then, we present our evaluation methodology and simulation results in Section 5 and 6 respectively. Finally, we conclude the paper in Section 7.

2 Problem Formulation

2.1 Primary Egress Point Selection Problem Formulation

We make the following assumptions prior to the problem formulation: (1) we focus the TE optimization objective only on inter-domain resources[1]. (2) we apply our work

[1] This assumption is according to the fact that capacity over-provisioning is usually employed by ISPs within their IP backbones [10].

Table 1. Notation Used In This Paper

Notation	Description
K	A set of destination prefixes, indexed by k
L	A set of egress points, indexed by l
S	A set of states $S=\{\varnothing\, U\, (\forall\, l\in\, L)\}$, indexed by s
I	A set of ingress points, indexed by i
$t(k,i)$	Bandwidth demand of traffic flows destined to destination prefix $k\in K$ at ingress point $i\in I$
$Out(k)$	A set of egress points that have reachability to destination prefix k
c^l_{inter}	Capacity of the egress point l
x^l_{sk}	A binary variable indicating whether prefix k is assigned to the egress point l in state s
u^l_s	Utilization on non-failed egress point l in state s. Its value is zero when $s=l$
$U_{max}(s)$	maximum egress point utilization in state s
U^{FS}_{Ave}	Average of maximum egress point utilization across all failure states

to the single egress selection case and on a general network model where each EP is composed of an egress router attached to a single inter-domain link[2].

In this section, we review the problem formulation of single egress selection described in [2]. This determines the PEPs under NS ($s=\varnothing$), and is hence the first level of our bi-level outbound TE problem. Table 1 shows the notation used in this paper.

Each element of the inter-domain TM, $t(k,i)$, represents the total volume of traffic from ingress point i towards destination prefix k. Due to the increasing use of multihoming, a prefix usually can be reached through multiple EPs, thereby allowing outbound TE to select the best PEP for the traffic. Given an inter-domain topology, destination prefixes and an inter-domain TM, the task of single egress selection is to determine the best PEP for each destination prefix[3]. The optimization objective we consider is to minimize the maximum EP utilization, which is defined as the highest utilization among all EPs:

$$Minimize\ U_{max}(\varnothing) = Minimize \underset{\forall l\in L}{Max}(u^l_{\varnothing}) = Minimize \underset{\forall l\in L}{Max}\left(\frac{\sum_{k\in K}\sum_{i\in I} x^l_{\varnothing k}t(k,i)}{c^l_{inter}}\right) \tag{1}$$

subject to the following constraints:

$$\forall k\in K: \sum_{l\in Out(k)} x^l_{\varnothing k} = 1 \tag{2}$$

$$\forall l\in L, k\in K: x^l_{\varnothing k}\in\{0,1\} \tag{3}$$

Minimizing objective function (1) ensures that traffic is moved away from congested to less utilized EPs and attempts to achieve load balancing across all EPs. Constraints

[2] In [2], outbound TE is divided into Single and Multiple Egress Selection. Since the objective of this paper is to demonstrate the principle of robust outbound TE, we consider assumption 2. Nevertheless, our idea is also applicable to multiple egress selection and multiple inter-domain links attached to each EP.

[3] Assigning a PEP to a destination prefix is equivalent to selecting that PEP for traffic demands that head towards that destination prefix.

(2) and (3) ensure that only one PEP is selected for each destination prefix under NS. In [2], an EP capacity constraint is added to the problem formulation. Nevertheless, we believe that our uncapacitated version is adequate since objective function (1) is effectively similar to the EP capacity constraint. The single egress selection problem has been proven to be NP-hard [2] by reducing it to the Generalized Assignment Problem (GAP), which is itself NP-hard.

2.2 Bi-level Egress Point Selection Problem Formulation

Given the inputs for the single egress selection, the goal of bi-level outbound TE is to determine, for each destination prefix, both a PEP under NS and a SEP that will serve the traffic when the PEP has failed (i.e. under FS). The optimization objective of the bi-level outbound TE problem is to minimize both the maximum EP utilization under NS and the average maximum EP utilization across all FSs. Recall that each FS corresponds to a single EP failure. The number of FSs is hence equal to the number of EPs |*L*|. By adding the NS, the total number of states |*S*| is |*L*| + *1*. The computational complexity of the bi-level outbound TE problem is thus an increasing function of the total number of states. To reduce this complexity, one may take the idea in [7] of performing the TE only on a small subset of FSs whose failures have significant impact on network performance. This set of EPs is referred to as *critical* EPs but we leave this as future work. The maximum EP utilization under FS *s* can be calculated in a similar way to (1) as:

$$\forall s \in S : Minimize\, U_{\max}(s) = Minimize \underset{\forall l \neq s}{Max}(u_s^l) = Minimize \underset{\forall l \neq s}{Max}\left(\frac{\sum_{k \in K}\sum_{i \in I} x_{sk}^l t(k,i)}{c_{\text{inter}}^l}\right) \tag{4}$$

The term $x_{sk}^l t(k,i)$ consists of flows which are assigned to EP *l* as their PEP and also flows which are assigned to EP *l* as their SEP. Since our optimization objective is to minimize the maximum EP utilization under both NS and FSs simultaneously, a bi-criteria optimization problem is formed. However, the two optimization objectives conflict with each other and hence we resort to a weighted sum approach to transform them into a single-criterion optimization problem, which is simpler to solve. The optimization objective function is thus:

$$Minimize\ F=(1\text{-}w)U_{max}(\varnothing)+wU_{Ave}^{FS}\ ,\ 0 \le w \le 1 \tag{5}$$

where

$$U_{Ave}^{FS} = \underset{\forall s \in S/\{\varnothing\}}{Ave}(U_{\max}(s)) = \frac{\sum_{s \in S/\{\varnothing\}} U_{\max}(s)}{|S|-1} \tag{6}$$

subject to the following constraints:

$$\forall k \in K, s \in S : \sum_{l \in Out(k)} x_{sk}^l = 1 \tag{7}$$

$$\forall l \in L, k \in K, s \in S : x_{sk}^l \in \{0,1\} \tag{8}$$

$$\forall l \in L, k \in K \quad if \quad x_{\varnothing k}^l = 1 \quad then \begin{Bmatrix} x_{sk}^l = 1 & \forall s \in S/\{l\} \\ x_{sk}^l = 0 & \forall s = l \end{Bmatrix} \tag{9}$$

By varying weight *w* and re-solving *F*, one can generate a trade-off curve between the two objectives using the weighting method of multi-objective programming [10]. If we solve the problem with *w=0*, the problem is simply reduced to the PEP selection problem. If *w=1*, the problem then completely ignores the performance under NS. In this paper, we present results for *w=0.5* (i.e. equal weight to the objectives optimized under NS and FS), which allows us to achieve significant performance improvement for SEP selection with only a small performance degradation for the PEP selection. Constraints (7) and (8) are equivalent to constraints (2) and (3), ensuring that only one EP is selected for each destination prefix as the PEP under NS ($s=\varnothing$) and only one EP is selected for each prefix as the SEP under FSs. Constraint (9) ensures that if prefix *k* is assigned to EP *l* in NS, then this prefix remains on *l* for all the FSs except when the current FS is the failure on *l*.

It is not surprising that the bi-level outbound TE problem is NP-hard, since it is an extension of the PEP selection problem, which is itself NP-hard. If the number of FSs is zero, the bi-level outbound TE is reduced to the PEP selection problem. As a result, we resort to using a heuristic approach to solve the problem.

For the implementation of the bi-level outbound TE solution, we can assign for each prefix the largest value of BGP *local-pref* for the selected PEP, the second largest value for the selected SEP and smaller values for the rest of the EPs. Whenever a PEP fails, the EP with the next largest *local-pref* (i.e. the SEP) becomes the exit point for the traffic headed towards the destinations. The solution can also be implemented by the proposal in [4] in which an IP tunnel is established to move traffic from the failed PEP to the precomputed SEP for faster failure recovery.

3 Proposed Tabu Search Heuristic

The Tabu Search (TS) methodology [8] guides local search methods to overcome local optimality and attempts to obtain near-optimal solutions for NP-hard optimization problems. Due to space limitations, the reader is referred to [8] for an overview of TS. In general, our proposed TS heuristic first requires initial PEP and SEP selection solutions, and then proceeds to obtain neighbor solutions by using a neighborhood search strategy in order to gradually enhance the quality of the initial solution.

3.1 Non-TE Initial Solution

We obtain initial PEP and SEP selection solutions by randomly selecting EPs for the destination prefixes while satisfying constraints (7) to (9). These initial solutions can be regarded as non-TE (i.e. non-optimized) solutions. The rationale of using such initial solutions is to demonstrate the effectiveness of the proposed TS heuristic in producing good performance from poorly performing initial solutions.

3.2 Neighborhood Search Strategy

A *move* transforms the current (initial) solution into a neighbor solution. To perform a move, we apply the SUBROUTINE_BESTMOVE heuristic shown in Figure 1, to first identify the best move for each FS and then select the best one among all the FSs.

```
SubRoutine_BestMove:
1.  For each $s \in S/\{\varnothing\}$
2.     Store the $PEP_{current}$, $current\_cost \leftarrow (1-w)U_{max}(\varnothing)+wU_{max}(s)$ and $j \leftarrow 0$
3.     For each $k \in MUEP_s$
4.        temporarily shift k from $MUEP_s$ to $LUEP_s$ to achieve the new solution $PEP_{new}$
5.        call SubRoutine_Greedy_Heuristic for state $s$ and temporarily make changes for current SEP
6.        $new\_cost \leftarrow (1-w)U'_{max}(\varnothing)+wU'_{max}(s)$ and $j \leftarrow j+1$
7.        $diff(j) \leftarrow current\_cost - new\_cost$ and restore the $PEP_{current}$
8.     find $\underset{j}{Max}\, diff(j)$ and its coresponding $PEP_{new}$, $PEP_{state\_best} \leftarrow PEP_{new}$// the best move for each FS
9.  For each $s \in S/\{\varnothing\}$
10.    temporarily implement the current $PEP_{state\_best}$
11.    call SubRoutine_Greedy_Heuristic for all FSs to achieve $SEP_{state\_best}$, implement it temporarily
12.    calculate $F=(1-w)U_{max}(\varnothing)+wU^{FS}_{Ave}$
13. Find Minimum $F$   // to find the best move among all the FSs ($PEP_{state\_best}$,$SEP_{state\_best}$)
14. Accept the changes that yield the Minimum $F$
```

Fig. 1. SubRoutine_BestMove

The following steps explain how to identify the best move for each FS:

Step 1. Store the currently assigned PEP for all prefixes in $PEP_{current}$. Calculate the *current_cost*, i.e. the weighted sum of the maximum EP utilization under both NS and the current FS (Figure 1 line 2). List all the prefixes in $PEP_{current}$ assigned to the Most Utilized EP under the current FS ($MUEP_s$)[4]. Consider each prefix at a time in the list and apply steps 2 to 4 until all the destination prefixes in the list have been considered (Figure 1 lines 3 to 7).

Step 2. Shift the prefix's PEP from $MUEP_s$ to the Least Utilized EP ($LUEP_s$)[5] (the goal of this move is to attract traffic towards the $LUEP_s$ and potentially to reduce the load on the $MUEP_s$). This results in a new solution for the PEP selection, which is denoted by PEP_{new}.

Step 3. Reassign the SEPs for the destination prefixes that have been assigned to the failed EP by using the SubRoutine_Greedy_Heuristic algorithm. The algorithm works as follows: (a) Sort all the destination prefixes on the failed EP by descending volume of traffic. (b) Take the first of these ordered prefixes and select as its SEP the available EP with the minimum utilization. (c) Repeat step (b) for the rest of the destination prefixes in order.

Step 4. Calculate the *new_cost* in the same way as the *current_cost* for the latest solution (Figure 1 line 6). Then calculate the difference between the *current_cost* and *new_cost* (i.e. *diff = current_cost- new_cost*). Restore the $PEP_{current}$.

[4] $MUEP_s$ is the link that has $\underset{\forall l \neq s}{Max}\, u_s^l$.

[5] $LUEP_s$ is the link that has $\underset{\forall l \neq s}{Min}\, u_s^l$.

Step 5. Identify the prefix that produces the largest value of *diff* (i.e. largest difference between the *current_cost* and *new_cost*). Consider the PEP_{new} that corresponds to this prefix as the best move for the current FS. Store this PEP_{new} in PEP_{state_best}.
Step 6.Repeat steps 1 to 5 for each FS and identify their PEP_{state_best} until all the FSs have been considered (Figure 1 lines 1 to 8).

After identifying the best move for each FS, we now identify the best of the best moves for all FSs by the following steps:

Step 1. For the best move for each FS, reassign the SEPs (SEP_{state_best}) for the corresponding PEP_{state_best} by using the **SubRoutine_Greedy_Heuristic** algorithm for all the FSs. (this calls the subroutine *s* times, once for each FS). Calculate objective function (5). Repeat step 1 for the best move of the next FS until all the FSs have been considered (Figure 1 lines 9 to 12).
Step 2. For all the FSs evaluated in step 1, choose the best move (i.e the PEP_{state_best} and its corresponding SEP_{state_best}) that yields the minimum objective value (Figure 1 lines 13-14).

3.3 Tabu List

The tabu list is a memory list that memorizes the most recent moves, operating as a first-in-first-out queue. As suggested in [8], the size of the tabu list depends on the size and characteristics of the problem. Since in our algorithm the attributes of a move are MUEP, LUEP and shifted destination prefixes, the size of the tabu list is determined by the number of destination prefixes. We define the size of the tabu list to be *total number of destination prefixes* / $|L|$.

3.4 Diversification

The goal of diversification is to prevent the searching procedure from indefinitely exploring a region of the solution space that consists of only poor quality solutions. It is a modification of the neighbourhood searching strategy and is applied when there is no obvious performance improvement after a certain number of iterations. For diversification, a group of highly and lightly utilized EPs are chosen for shifting destination prefixes under a FS. We define the threshold of obvious performance improvement to be 10% of the best visited solution and the number of iterations to be 10% of the maximum iteration mentioned below.

3.5 Stopping Criterion

Many stopping criteria can be developed depending on the nature of the problem. The most common criterion, used in this paper, is to define a maximum number of iterations. However, we do not arbitrary select the number of maximum iterations since the performance of the TS heuristic mainly depends on how many times the PEPs and SEPs are reassigned. We found that setting the maximum iteration number to be 5 times the number of destination prefixes gives us sufficiently good results.

4 Alternative Strategies

Our proposed TS heuristic is only one of several approaches in solving the bi-level outbound TE problem. In this section, we present three alternative approaches. For these approaches, optimal-aware heuristic is used for the PEP selection and the three alterative approaches only differ in their SEP selection. We remark that the optimal-aware heuristic is our best attempt in solving our PEP selection problem, as no algorithm for solving the problem with objective function (1) has been proposed in the literature. The optimal-aware heuristic works as follows:

Step 1**:** Calculate the mean utilization by dividing the total traffic volume by the total capacity of all EPs. We regard this mean utilization as the theoretical optimal (i.e. the most load balanced) utilization targeted for each EP to achieve. However, this theoretical result is not a valid solution because it allows arbitrary traffic splitting over any EP, violating constraints (7) and (8). Nevertheless, it is used as an "NS lower bound" solution[6] for comparing performance with other strategies.
Step 2**:** To ensure that each EP does not exceed the theoretical optimal utilization, set the mean utilization as a capacity constraint on each EP.
Step 3**:** Sort the destination prefixes in descending order according to the amount of traffic they carry and choose one at a time in order.
Step 4**:** Select the EP with the minimum utilization as the PEP of this destination prefix if it satisfies the capacity constraint, if not proceed to the next prefix. Repeat this step until all the destination prefixes have been considered.
Step 5**:** If there exist unassigned destination prefixes because of capacity constraint violation, re-run step 4 without considering the capacity constraint.

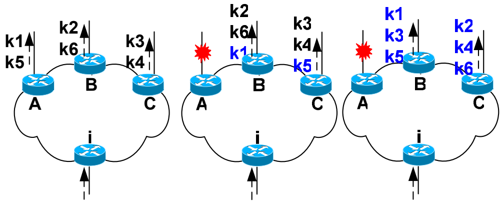
4.1 Random Reassignment Strategy

In the Random Reassignment (**RandomR**) strategy, when an EP fails, the prefixes on the failed EP are re-assigned to other available but *randomly* chosen EPs. We illustrate an example of the **RandomR** in Figure 2. In this example there are three EPs (A, B and C) with equal capacity (60Mbps) and an ingress point *i*. The input traffic flows and their traffic volume are shown in Table 2. Figure 2(a) shows a solution of the PEP selection, which can be generated by the optimal-aware heuristic. The solution has the best load balancing over all the EPs. Figure 2(b) shows the solution of the SEP selection under EP A failure produced by the **RandomR.** The figure demonstrates that when EP A is assumed to fail, destination prefixes *k1* and *k5* are then randomly assigned to EP B and C respectively as their SEPs. This random assignment, however, causes heavy load on EP B which could easily lead to congestion (e.g. $u_A^B = \frac{20+20+10}{60} = 0.833$, $u_A^C = \frac{10+10+10}{60} = 0.5$). Therefore, the **RandomR** performs poorly under any FS since during optimization failures are not taken into account. Nevertheless, since only the affected destination prefixes are reassigned, the level of traffic disruption is minimized (i.e. only prefixes *k1* and *k5* are disrupted when EP A fails).

[6] In a similar fashion, we define the "FS lower bound" to be the total volume of traffic divided by the capacity of all EPs excluding the failed one.

Table 2. Input Traffic Flows

TRAFFIC FLOW	TRAFFIC VOLUME(MBPS)
t(k1,i)	20
t(k2,i)	20
t(k3,i)	10
t(k4,i)	10
t(k5,i)	10
t(k6,i)	10



2(a) 2(b) 2(c)

Fig. 2. Show the destination prefix assignment according to (a) **OPTIMAL-AWARE HEURISTIC** , (b) **RANDOMR** if EP A fails and (c) **GLOBALR** if EP A fails

4.2 Global Reassignment Strategy

In the Global Reassignment (**GLOBALR**) strategy, for any EP failure, the **OPTIMAL-AWARE HEURISTIC** is reapplied to perform PEP selection from scratch. Such network-wide computation can be regarded as the best approach with respect to performance but possible large traffic disruption because the PEPs for most of destination prefixes are likely changed. We use the **GLOBALR** as a benchmark for comparing the performance to other strategies. Figure 2(c) shows the result of the **GLOBALR** based on the previous example. As can be seen, when EP A fails, some prefixes are reassigned away from their original EPs. For example, *k2* and *k6* are shifted from EP B to C while *k3* is shifted from EP C to B. Nevertheless, the utilization upon any EP failure is optimal (i.e. $u_A^B = \frac{20+10+10}{60} = 0.666, u_A^C = \frac{20+10+10}{60} = 0.666$).

4.3 Greedy Reassignment Strategy

In the Greedy Reassignment (**GREEDYR**) strategy, for any EP failure, only the prefixes assigned on the failed EP are re-assigned by a greedy heuristic as follows: the prefix that carries the largest amount of traffic is reassigned to the available EP that has the minimum utilization. This step repeats for the rest of the affected prefixes.

5 Evaluation Methodology

5.1 Network Topology and Inter-domain Traffic Matrices

Our experiment is performed on topologies with 3, 6 and 10 EPs. We note that the 3-EP topology is the smallest scenario where the bi-level outbound TE is applicable. Larger topologies then evaluate performance scalability of the proposed strategies.

We assume the capacity of all the EPs to be OC-12 (622Mbps). For scalability and stability concerns, outbound TE can focus only on a small fraction of Internet destination prefixes, which are responsible for a large fraction of the traffic [1]. In this paper, we consider 30, 60 and 100 such popular destination prefixes for 3, 6 and 10-EP topologies respectively. In fact, each of them may not merely represent an individual prefix but also a group of distinct destination prefixes that have the same set of candidate EPs [12] in order to improve network and TE algorithm scalability. Hence, the number of prefixes we consider could actually represent an even larger value of actual prefixes. Without loss of generality, we assume that each EP has reachability to all the considered destination prefixes.

We generate synthetic traffic matrices for our evaluation. We generate inter-domain traffic from each ingress point towards each of the considered destination prefixes. Previous work has shown that inter-domain traffic is not uniformly distributed [11]. According to [12], the volume of inter-domain traffic demand is top-heavy and it can be approximated by Weibull distribution with the shape parameter equal to 0.2-0.3. We generate the inter-domain TM following this distribution with the shape parameter equal to 0.2. We remark that our TM generation process is just our best attempt to model inter-domain traffic, as no synthetic model for the actual behavior of traffic in real networks can be found in the literature.

5.2 Performance Matrices

The following performance metrics are used to evaluate the proposed strategies. For these metrics, lower values are better than high values.

- ***NS maximum EP utilization:*** this refers to $U_{max}(\varnothing)$.
- ***Average of maximum EP utilization across all FSs***: this refers to U_{Ave}^{FS} .
- ***Percentage of the average disrupted traffic volume:*** a traffic flow is disrupted if it is shifted to another EP when a failure occurs. We denote the volume of disrupted traffic under FS s by DT_s and the average of disrupted traffic volume by *AveDT*. The percentage of the average disrupted traffic volume (*PerAveDT*) is the ratio of the average disrupted traffic volume to the total traffic volume (|*T*|):

$$PerAveDT = \frac{AveDT}{|T|} \times 100 = \frac{\sum_{s=1}^{|S|} DT_s \Big/ |S|-1}{|T|} \times 100 \quad \text{where } |T| = \sum_{k \in K} \sum_{i \in I} t(k,i) \tag{10}$$

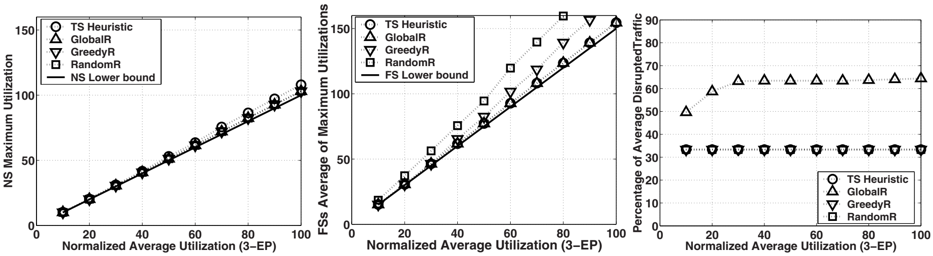
6 Simulation Results

6.1 Evaluation of Normal State Maximum EP Utilization

Figure 3(a) shows the NS maximum EP utilization achieved by different strategies for the 3-EP topology. The x-axis represents the normalized average utilization which is the total traffic volume normalized by the total capacities of all EPs. All the simulation results presented in this paper are the average of 20 trials.

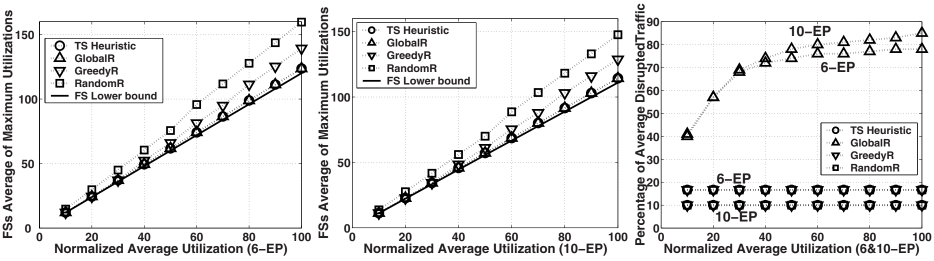
First of all, we can observe from the figure that the performances achieved by **RandomR**, **GlobalR** and **GreedyR** are identical. This phenomenon is expected since they use the same algorithm (optimal-aware heuristic) for their PEPs selection. The optimal-aware heuristic produces near-optimal performance that is only within 1%-3% from the NS lower bound[7]. On the other hand, however, the TS heuristic has

[7] Obviously the NS lower bound curve is linear because it is equal to the normalized average EP utilization.



3(a) 3(b) 3(c)

Fig. 3. Performance evaluation for 3-EP topology



4(a) 4(b) 4(c)

Fig. 4. Performance evaluation for 6 and 10-EP topology

slightly higher maximum EP utilization than the others (about 1%-5% compared to **GLOBALR** and 2%-8% compared to the NS lower bound). This can be explained by the reason that the TS heuristic attempts to minimize the maximum EP utilization under both NS and FSs simultaneously, as shown by objective function (5). Since the two objectives do not coincide, there is a performance trade-off between them. Nevertheless, as will be shown next, the TS heuristic significantly improves the performance across FSs at the cost of only a small performance degradation under NS.

6.2 Evaluation of the Average Maximum EP Utilization Across all Failure States

Figure 3(b)[8] shows the average of maximum EP utilization across all FSs achieved by different strategies for the 3-EP topology. The figure shows that the TS heuristic and **GLOBALR** have similar results and are within 1%-3% of the FS lower bound. The reason is that both strategies have given attention to optimizing the performance under FSs (i.e. by using objective function (5) for the TS heuristic and EP selection re-computation under each FS for the **GLOBALR**). On the other hand, for the **GREEDYR**, the performance degrades 1%-13% from the **GLOBALR** and the TS heuristic and 2%-16% from the FS lower bound. This performance degradation is expected since

[8] For completeness we show performance results for two scenarios, one for below 100% maximum utilization and the other one for over 100% maximum utilization.

the GreedyR only considers minimizing EP utilization under FSs as the second optimization objective. In other words, the performance objectives under NS and FSs are optimized in a lexicographic importance order. As a result, the performance as measured by maximum EP utilization under FSs are not truly optimized: the solution of the PEP may not be a good input for GreedyR to produce the optimal SEP. In addition, the GreedyR performance starts to degrade compared to the TS heuristic and GlobalR when the normalized average utilization (x-axis) exceeds 30%. In fact, with lower normalized average utilization, the EPs comfortably have extra capacity to accommodate the other traffic flows assigned by the GreedyR and keep the utilization balanced. However, as the normalized average utilization increases, the residual capacity of EPs reduces and the PEP solution restricts the ability of the GreedyR to reassign the prefixes of flows from the failed EP. Finally the RandomR has dramatic performance degradation, being about 22%-30% worsen than the GlobalR and the TS heuristic. This performance degradation is primary due to the random SEP selection, which does not optimize any performance objective.

6.3 Evaluation of the Average Disrupted Traffic Volume

Figure 3(c) presents the percentage of the average disrupted traffic volume for the 3-EP topology. The figure shows that the TS heuristic, RandomR and GreedyR have identical and constant performance as the normalized average utilization increases. This is due to the fact that with all these strategies, only the traffic on the failed EP is shifted. With the 3-EP topology, the number of FSs is 3 and this results in minimum (1/3)*100=33% traffic disruption for any single EP failure. However, since the GlobalR performs network-wide recomputation for any single EP failure, both the affected and unaffected destination prefixes are likely to be reassigned, thereby causing significant traffic disruption in particular when the normalized average utilization is high. The figure shows that the average disrupted traffic volume for the TS heuristic, RandomR and GreedyR are 33%-48% better than the GlobalR.

6.4 Evaluation of Larger Topologies

We also performed our evaluation on larger topologies with 6 and 10 EPs. We note that the result patterns of NS maximum EP utilization for the 6 and 10-EP topologies are similar to those in Figure 3(a), hence, we proceed our performance analysis in a similar fashion to that in Section 6.1. Figures 4(a) and (b) present the average of maximum EP utilization across all the FSs for 6 and 10-EP topologies respectively. We observe that the higher the total number of EPs the lower the average of maximum EP utilization. As with the 3-EP topology, we can reach a conclusion: the TS heuristic always performs as well as the GlobalR and better than the others and is very closer to the FS lower bound. This shows that the performance achieved by the TS heuristic scales well for larger topologies.

Figure 4(c) presents the percentage of average disrupted traffic volume for the 6 and 10-EP topologies together. By comparing Figure 3(c) with 4(c), we observe that, as the number of EPs increases, the percentage of the average disrupted traffic volume for the TS heuristic, RandomR and GreedyR decreases. Conversely, as the number of EPs increases, this performance metric for the GlobalR increases. This is attributed to the fact that, by increasing the number of EPs, the solution spaces for the GLOBALR's prefix reassignment is greatly enlarged. As a result, the likelihood that the prefixes changes from the originally assigned EPs to other EPs increases.

6.5 Overall Performance

In summary, our proposed TS heuristic regarding the average of maximum EP utilization across FSs performs (1) as well as the GLOBALR, (2) almost as well as the FS lower bound, (3) better than the GREEDYR when the normalized average utilization exceeds 30%, and (4) always significantly better than the RANDOMR. The excellent performance of the TS heuristic under FS is only at the cost of a small performance degradation in the NS maximum EP utilization compared to the other strategies. The TS heuristic also keeps the traffic disruption to a minimum. Hence, overall, it can be regarded as the best among all the strategies.

The GLOBALR performs almost as well as the NS and FS lower bounds. However, it causes very large traffic disruption which leads to frequent BGP configuration changes and route instability. Hence, it is an impractical strategy. The GREEDYR performs as well as the GLOBALR and almost as well as the NS lower bound but it has significant performance degradation in the average of maximum EP utilization across all FSs as the normalized average utilization increases. Finally the RANDOMR is the worst performer in the average of maximum EP utilization across all FSs, which makes it inappropriate for robust TE.

7 Conclusion

In this paper, we have proposed a bi-level outbound TE optimization approach to make outbound TE robust to EP failures. This approach determines for each destination prefix the best PEP and the SEP (the next best EP) upon PEP failure. The optimization objectives are to minimize the maximum EP utilization under NS and the average of maximum EP utilization across all FSs simultaneously. We have proposed a tabu search heuristic to solve the problem and compared its performance to three alternative approaches. Our simulation results show that the tabu search heuristic significantly reduces the average of maximum EP utilization across all the FSs at a cost of only small increases in the NS maximum EP utilization. It also keeps the traffic disruption to a minimum. The other alternative approaches, however, do not achieve all these objectives together. We believe that our work provides insights to network operators on how to make optimal BGP outbound route selection robust to inter-domain EP failures which are common events and transient in nature.

References

1. N. Feamster et al.: Guidelines for Interdomain Traffic Engineering, *ACM CCR*, 2003.
2. B. Bressound et al.: Optimal Configuration for BGP Route Selection, *INFOCOM*, 2003.
3. K. Ho et al.: Multi-objective Egress Router Selection Policies for Inter-domain Traffic with Bandwidth Guarantees, *IFIP Networking*, 2004.

4. O. Bonaventure et al.: Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures, *ACM CONEXT*, 2005.
5. B. Fortz et al.: Optimizing OSPF/IS-IS Weights in a Changing World, *IEEE JSAC*, 2002.
6. A. Nucci et al.: IGP Link Weight Assignment for Transit Link Failures, *ITC*, 2003.
7. A. Sridharan et al.: Making IGP Routing Robust to Link Failures, *IFIP Networking*, 2005.
8. F. Golver et al. *Tabu Search.* Kluwer Academic Publisher, Norwell MA 1997.
9. T. Telkamp: Traffic Characteristics and Network Planning, *NANOG 2002*.
10. J.L. Cohon. *Multiobjective Programming and Planning*. Academic Press, New York 1978.
11. W. Fang et al.: Inter-AS Traffic Patterns and their Implications, *IEEE GLOBECOM*, 1998.
12. A. Broido et al.: Their Shares: Diversity and Disparity in IP Traffic, *PAM*, 2004.

An Approach to Off-Line Inter-domain QoS-Aware Resource Optimization

Manuel Pedro[1,2], Edmundo Monteiro[2], and Fernando Boavida[2]

[1] Polytechnic Institute of Leiria, School of Technology and Management,
Morro do Lena, Alto do Vieiro, 2411-901 Leiria, Portugal
[2] University of Coimbra, Pólo II, Pinhal de Marrocos,
3030-290 Coimbra, Portugal
{macpedro, edmundo, boavida}@dei.uc.pt

Abstract. Inter-domain traffic engineering is a key issue when QoS-aware resource optimization is concerned. Mapping inter-domain traffic flows into existing service level agreements is, in general, a complex problem, for which some algorithms have recently been proposed in the literature. In this paper a modified version of a multi-objective genetic algorithm is proposed, in order to optimize the utilization of domain resources from several perspectives: bandwidth, monetary cost, and routing trustworthiness. Results show trade-off solutions and "optimal" solutions for each perspective. The proposal is a useful tool in inter-domain management because it can assist and simplify the decision process.

1 Introduction

The main purpose of inter-domain resource optimization is to map incoming inter-domain traffic flows into inter-domain network resources, satisfying quality of service (QoS) requirements, while aiming at optimizing the use of network resources across autonomous systems (AS) boundaries. Network resources usage is, in any case, conditioned by existing Service Level Specifications (SLSs) that, in turn, result from the Service Level Agreements (SLAs) established between each domain and its neighbors. For the purpose of this paper, the terms 'domain' and 'autonomous system' are synonyms.

In order to describe the inter-domain relationships of an autonomous system, one can use a simple model, as shown in Fig. 1. An autonomous system is interconnected with other autonomous systems by means of its ingress and egress interfaces. For the propose of this paper, the terms 'interfaces' and 'links' are synonymous.

The service offerings between autonomous systems as well as their mutual responsibilities are described by means of Service Level Agreements. In general, each SLA defines a set of contractual, administrative and technical requirements. The latter are called Service Level Specifications. An SLS comprises several items or clauses, including identification, application scope, flow identification, traffic conformance, excess treatment, and performance guarantees.

In the context of the present work an SLS is characterized by an egress interface, an inter-domain QoS class q as proposed in [7], a destination prefix d, the corresponding

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 247 – 255, 2006.

maximum bandwidth requirements *b*, the monetary cost per unit of bandwidth *c* , and the route trustworthiness *r* associated to the SLS. The monetary cost component reflects the monetary cost associated with the established SLA. On the other hand the routing trustworthiness reflects the intra-domain routing costs associated with the egress interface, and the inter-domain routing costs like route quality, reliability and domain policies. An SLS entry for a domain has the following format:

```
SLS entry = [egress interface, q, d, b, c, r]
```

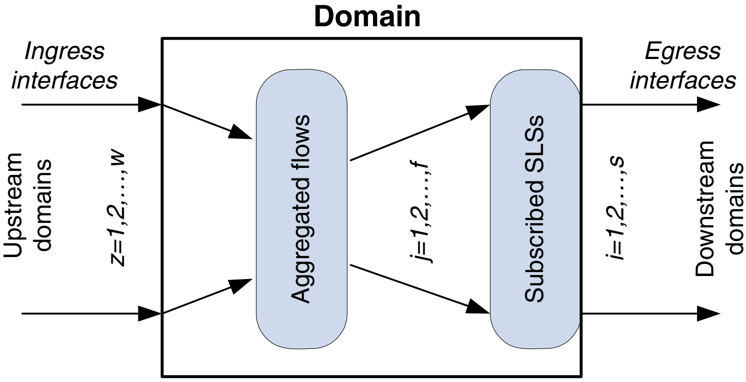


Fig. 1. Inter-domain relationship model

On the other hand, a domain receives from upstream domains a collection of *w* data flows towards other domains. Depending on the domain policy and on their common characteristics, such as destination and QoS class, these flows may be aggregated into *f* inter-domain traffic flows. The flows' common characterization includes the inter-domain class mapping *q* and the destination prefix *d*. That is, an aggregated flow entry has the following format:

```
Aggregate flow entry = [ingress interface, q, d, a]
```

where *a* is the bandwidth requirement of the aggregated flow. The flow will be mapped into one of the existing SLSs. The appropriate selection of the SLSs for the inter-domain traffic flows benefits the domain by improving the network resources, maximizes the profits [21] from a business point-of-view and, at same time, selects the most reliable routes according to internal and external information and business objectives. The first benefit is reached through a correct bandwidth load-balancing, the second through a minimization of the costs, and the third through a high value of routing trustworthiness. In contrast, in current networks this task is executed in a trial-and-error fashion.

The problem can be expressed by three objective functions (1), (2), and (3) that represent respectively the total costs for bandwidth, monetary cost and routing. Formally, the problem can be stated as follows. Let $I = \{1,2,...,s\}$ be the set of SLSs and $J = \{1,2,...,f\}$ the set of aggregated traffic flows. For each SLS *i* there is a given resource capacity, expressed in terms of bandwidth, $b_i > 0$. For each $i \in I$ and each $j \in J$ there is a given set of costs, $B_{i,j} > 0$, for bandwidth, $C_{i,j} > 0$, for monetary, and $R_i > 0$, for routing, for assigning an aggregated traffic flow *j* to an SLS *i*. Additionally, $z_{i,j}$ is an indicator function that returns 1 if the traffic flow *j* is assigned to SLS *i* and 0 otherwise. The mathematical formulation is as follows:

$$y_1 = \sum_{i=1}^{s}\sum_{j=1}^{f} B_{i,j} \cdot z_{i,j} \tag{1}$$

$$y_2 = \sum_{i=1}^{s}\sum_{j=1}^{f} C_{i,j} \cdot z_{i,j} \tag{2}$$

$$y_3 = \sum_{i=1}^{s}\sum_{j=1}^{f} R_i \cdot z_{i,j} \tag{3}$$

$$\text{subject to } \sum_{j=1}^{f} a_{i,j} \cdot z_{i,j} \le b_i, \forall i \in I, \tag{4}$$

$$\text{with } \sum_{i=1}^{s} z_{i,j} = 1, \forall j \in J, \tag{5}$$

$$z_{i,j} \in \{0,1\}, \forall i \in I, \forall j \in J. \tag{6}$$

The goal is to minimize the costs of (1), (2), and (3), where $B_{i,j}$, $C_{i,j}$, and R_i are respectively the cost functions for bandwidth, monetary, and routing as described in Sec. 3.1. The capacity constraint (4) ensures that the total resource requirements of the traffic flows assigned to each SLS do not exceed the available capacity. The assignment constraint (5) guarantees that each traffic flow is assigned to exactly one SLS.

Since we need to find the best solution considering all objectives at the same time, our problem falls into a multiple objective optimization problem. For this kind of problems genetic algorithms are a well known technique capable of finding the entire non-dominated Pareto front in a single run [1]. A Pareto front is a set of solutions, and a solution is said to be non-dominated if its components cannot be improved in terms of one objective without causing a simultaneous degradation in at least one of the other components [9]. The minimization problem is expressed formally as follows, for *n* objective functions with *m* optimization parameters:

$$\text{Minimize } y = f(x) = (f_1(x),...,f_n(x)) = (y_1,..., y_n) \in Y \tag{7}$$

$$\text{and } x = (x_1,..., x_m) \in X \tag{8}$$

With X as the parameter space and Y the objective space, x is called the decision vector and y the objective vector. A decision vector $a \in X$ is said to dominate a decision vector $b \in X$ if and only if:

$$\forall i \in \{1,...,n\} : f_i(a) \geq f_i(b) \wedge \exists j \in \{1,...,n\} : f_j(a) > f_j(b) \tag{9}$$

The objective of the work presented in this paper is to propose the optimization of domain resources from multiple perspectives, namely from the bandwidth usage perspective, monetary cost perspective, in line with [21], and routing costs. For this, three objective functions are proposed, and our aim is supported by a multiple-objectives evolutionary algorithm especially designed to deal with these off-line interdomain traffic engineering issues.

In Section 2 of this paper an overview of related work is given. This is followed by a presentation of the proposed objective functions and the supported evolutionary algorithm in Section 3. In order to validate our work, an evaluation framework comprising two test scenarios, each one representing a type of Internet transit autonomous system [4], is presented in Section 4. Section 5 presents and discusses the obtained results. The conclusions and guidelines for further work are presented in Section 6.

2 Related Work

Several studies on intra-domain resource optimization, such as [12-15], can be found in the literature. In the case of inter-domain, references [8][16-19] constitute the framework for most of the current proposals.

Genetic algorithms have already been extensively used to solve network optimization problems [8][12-15][20][22]. These algorithms, belonging to the class of evolution strategies used in optimization, resemble the process of biological evolution, where each individual is described by its genetic code, called a chromosome. On the other hand each chromosome is composed of individual genes. In the problem in hand, a gene is the assignment of a single aggregate traffic flow to an SLS, and an individual (i.e., a chromosome) is a potential solution.

There are several examples of single objective proposals, like in [2] where different heuristic algorithms are compared, or in [6] where a weighted genetic algorithm is presented. Other proposals as in [22] address the minimization of inter-domain transit costs.

The proposal in [8] presents a multi-objective solution based on [9] that contemplates the cost minimization and the bandwidth cost minimization, for traffic engineering of best effort traffic. Our approach extends the use of routing trustworthiness costs for off-line traffic engineering.

To the best of the authors' knowledge, it is the first time an inter-domain optimization proposal includes simultaneous optimization of bandwidth costs, monetary costs, and routing costs. It is also the first time a parameter related to egress links and routing is used, combining internal, external, and business perspectives, which leads to a simplification of the egress link selection task. Lastly, a modified multi-objective genetic algorithm that simplifies the management task is proposed, allowing the choice of the perspective which best fits the objectives.

3 Proposal

This section describes the proposed cost functions and the evolutionary algorithm.

3.1 Cost functions

The off-line traffic engineering as considered in this paper consists of selecting the optimal mapping between sets of aggregated traffic flows and the associated sets of SLSs in such a way that the following the objectives are satisfied:

- minimization of the egress link bottleneck, thus improving the egress load-sharing
- minimization of the costs of egress links' usage, so as to maximize the domain business profit
- minimization of the routing costs, improving the link trustworthiness

The bandwidth objective function (10) was used in order to measure the egress interfaces bottleneck, allowing the correct load-balancing in these interfaces, where b_i is the available bandwidth on egress interface i (the agreed SLS) for some QoS class and destination and b_j the bandwidth of the aggregate flow j. The value 0.1 was added to the dominator in order to limit the values of $B_{i,j}$ to 100.

$$B_{i,j} = \frac{1}{(b_i - b_j + 0.1)^2} \tag{10}$$

$$C_{i,j} = c_i \cdot b_j \tag{11}$$

$$R_i = 100 - r_i \tag{12}$$

On the other hand, the monetary cost objective function (11) measures the charge to pay for using the established SLS i, by the aggregated flow j. It represents the domain expenses.

Lastly, the route trustworthiness objective function (12) measures aspects related to the egress links. This includes route fail history, link fail history, routing metrics history, intra-domain routing costs, and domain policies. The specific way to combine these data in order to obtain a value for the route trustworthiness is outside the scope of this paper. In this paper, this parameter takes values varying from 0 (link not used) to 100 (the best choice). The routing perspective is not a BGP weight, nor a routing metric, nor does it intend to replace the *local-pref* discretionary attribute [10].

3.2 The Algorithm

The proposed algorithm follows the proposal in [9]. The basic algorithm steps are presented in Fig. 2. It starts with the creation of the initial generation, where the individuals are created randomly. Then, an evaluating step based on the proposed objective functions (1),(2), and (3), with costs (10), (11), (12) respectively, follows. After that, and for a number of generations, a new generation of children is created that are compared with the corresponding generation of parents. From this comparison the better elements will compose the next generation of parents. The ranking step is done as proposed in [9].

```
Create the initial parent generation;
Evaluate the generation;
For a number of generations;
  Create the child generation;
  Evaluate both generations together;
  Rank both generations together;
  Replace worst parents with better children;
End
```

Fig. 2. Algorithm basic steps

The algorithm has a time complexity of $O(MN^2)$ where M is the number of objectives and N the size of the population.

4 Evaluation Framework

In order to evaluate our proposal, two typical scenarios were built, a tier-1 and a non-tier-1 transit autonomous' systems, as proposed in [4]. The scenarios' characterization is presented in Table 1. The traffic matrices and the SLSs matrices were built using these values as basis, Weibull distributions for sources and destination prefixes [5], Weibull distributions for ingress and egress links [3], and 3 exponentially distributed QoS classes [11].

For each scenario the algorithm returned a Pareto front with a non-dominated population of individuals ranked according. The comparison was made between front individual's rank 1 solution and the "best" ones in relation to every single perspective: bandwidth, monetary cost and routing cost.

Table 1. Test scenarios characterization

AS level	Tier 1	Non-Tier 1 transit
Links	400	20
QoS classes	3	3
Bandwidth (sources)	0..100	0..100
Destination prefixes	14738	14738
Monetary cost	1..10	1..10
Route trustworthiness	0..100	0..100
Aggregated flows	84400	50400

5 Results

The two scenarios presented in Table 1 were used for testing the assignment algorithm as described in Sec. 4. Fig. 3 shows the comparison between the 4 different solutions returned by the algorithm for the Tier 1 scenario, for each of costs. The figure shows the per cent increase in cost in relation to the minimum cost solution for the correspondent perspective (with 0%).

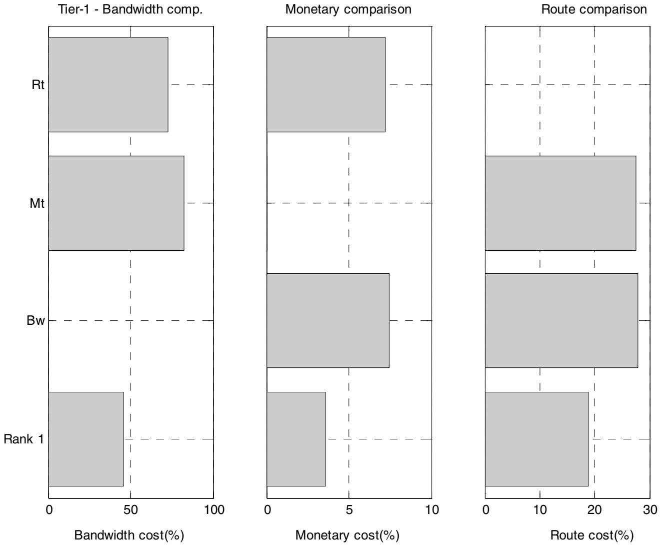


Fig. 3. Tier 1 results – comparison in percentage with lowest cost solution for bandwidth costs (*left*), monetary costs (*middle*), and routing costs (*right*)

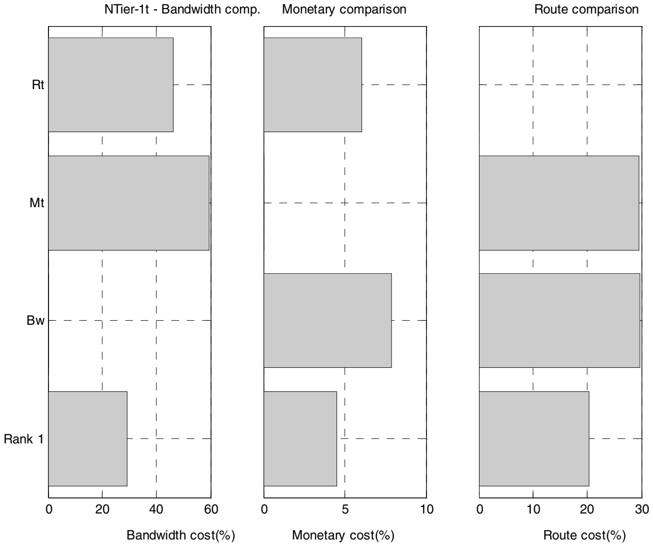


Fig. 4. Non-Tier 1 transit results –comparison in percentage with lowest cost solution, for bandwidth costs (*left*), monetary costs (*middle*), and routing costs (*right*)

When only the "best" bandwidth perspective is selected (*row two from the bottom in all graphics*) we have got the lowest costs for bandwidth but higher costs for monetary and route perspectives. On the other hand, if the selected individuals are the "best" from the monetary point-of-view (*row three from the bottom in all graphics*) we have got the lowest monetary cost, but high costs for bandwidth and route perspectives. The same we can say for the "route" perspective (*top row in all graphics*).

The bottom row in all graphics shows the individuals in the first rank, as returned by the algorithm. In this case the solution is not as good as when a perspective is selected individually, but has better costs compared with the worst cost values returned by the other solutions.

In the Fig. 4 a comparison between the four different solutions for the non-Tier 1 transit scenario, for each of the costs perspectives, is presented. The figure shows the per cent increase in cost in relation to the minimum cost solution for the correspondent perspective (with 0%). Comparatively, the results are similar to Tier 1.

6 Conclusions

Inter-domain QoS-aware resource optimization is one of the main challenges of current traffic engineering. Based on a modified version of a multi-objective genetic algorithm [9] and using typical models of transit autonomous systems, our proposal presents a set of solutions. They include trade-off solutions and "best" solutions from single perspectives optimizations. These solutions can be used to generate domain policies that, in turn, may influence routing decisions.

This paper also introduces a new kind of traffic engineering factor that simplifies the selection of the egress links: the route trustworthiness.

As a general conclusion, one can say that the presented proposal can be a useful tool in domain management because it simplifies the decision process by presenting the optimal costs either in terms of individual perspective, or in terms of trade-off between all perspectives.

Further work will address the algorithm's refinement and efficiency (the latter with the objective of reducing the computational complexity) and methods to compute route trustworthiness.

Acknowledgement

This work was partially supported by the Portuguese Ministry of Science and High Education (MCES) and by European Union FEDER under programs POSC and PRODEP.

References

1. Coello, C.: An Updated Survey of GA-Based Multiobjective Optimization Techniques. ACM Computing Surveys, 32(2) (2000) 109-143
2. Pedro, M., Monteiro, E., Boavida, F.: Comparative Study of Inter-Domain Traffic Optimization Algorithms. In: Proc. of IPS-MoMe 2005, Warsow, Poland (2005)

3. Fang, W., Peterson, L.: Inter-AS traffic patterns and their implications. In: GLOBECOM 1999 - IEEE Global Telecommunications Conference (1999) 1859-1868
4. Zhang, B., Liu, R., Massey, D., Zhang, L.: Collecting the Internet AS-level Topology. ACM SIGCOMM Computer Communication Review (CCR), special issue on Internet Vital Statistics (2005)
5. Broido, A., Hyun, Y., Gao, R., Claffy, K.: Their share: diversity and disparity in IP traffic. In: Proc. Passive and Active Network Measurement (2004)
6. Pedro, M., Monteiro, E., Boavida, F.: A Two-Phase Algorithm for Off-line Inter-domain Traffic Optimization. In: Proc. of International Conference on Service Assurance with Partial and Intermittent Resources (SAPIR 2005), Lisbon, Portugal (2005)
7. Levis, P. et al: A New Perspective for a Global QoS-based Internet. In the Journal of Communications Software and Systems (In press), Available online (www.mescal.org) (2005)
8. Uhlig, S.: A multiple-objectives evolutionary perspective to interdomain traffic engineering. International Journal of Computational Intelligence and Applications, Special Issue on Nature-Inspired Approaches to Telecommunications, World Scientific Publisher (2005)
9. Deb, K., Pratap. A, Agarwal, S., and Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation, 6(2), (2002). 181-197
10. Halabi, S., McPherson, D.: Internet Routing Architectures. 2nd edn. Cisco Press (2000)
11. Chang, W., Simon, R.: Performance Analysis for Multi-Service Networks with Congestion-Based Pricing for QoS Traffic. Annual Simulation Symposium (2005) 33-40
12. Ericsson, M., Resende, M., Pardalos, P.: A genetic algorithm for the weight setting problem in OSPF routing. Technical report, ATT Shannon Laboratory (2001)
13. Buriol, L., Resende, M., Ribeiro, C., Thorup, M.: A memetic algorithms for OSPF routing. In: Proceedings of 6th INFORMS Telecom (2002) 187—188
14. Buriol, L., Resende, M., Ribeiro, C., Thorup, M.: A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. Unpublished [Online]. Available: www.optimization-online.org/DB_FILE/2003/06/674.pdf (2003)
15. Riedl, A.: A hybrid genetic algorithm for routing optimization in IP networks utilizing bandwidth and delay metrics. In: Proceedings of IEEE Workshop on IP Operations and Management (IPOM), Dallas (2002)
16. Morand, P., et al.: D1.1: Specification of Business Models and a Functional Architecture for Inter-domain QoS Delivery. IST-2001-37961, unpublished. (2003)
17. Ho, K., Wang, N., Trimintzios, P., Pavlou, G., Howarth, M.: On Egress Router Selection for Inter-domain Traffic with Bandwidth Guarantees. In: Proceedings of IEEE Workshop in High Performance Switching and Routing (HPSR'2004), Phoenix, Arizona, USA (2004)
18. MESCAL project [website]: www.mescal.org
19. Bressoud, T., Rastogi, R., Smith, M.: Optimal Configuration for BGP Route Selection. In: Proceedings of IEEE INFOCOM' 2003, San Francisco (2003)
20. Pioro, M., and Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufmann Series in Networking (2004)
21. Aiber, S. et al.: Autonomic Self-Optimization According to Business Objectives. In: Proceedings of the International Conference on Autonomic Computing (ICAC'04) (2004)
22. Ho, R., Pavlou, G., Howarth, M., Wang, N.: An Incentive-based Quality of Service Aware Algorithm for Inter-AS Traffic Engineering. In: Proceedings of the IEEE International Workshop on IP Operations and Management (IPOM'2004), Beijing, China (2004)

A Distributed QoS Scheduler for Smoothing Output Traffic of Input Buffered Switches*

Man-Ting Choy and Tony T. Lee

Department of Information Engineering,
The Chinese University of Hong Kong
{mtchoy1, ttlee}@ie.cuhk.edu.hk
http://bblab.ie.cuhk.edu.hk/index.html

Abstract. To provide stringent service guarantees such as latency and backlog bounds for input-buffered switches, a set of scheduling algorithm and admission control strategy is proposed. This set of traffic control strategy is primarily based on a single-server scheduling algorithm called Smoothed Round Robin (SRR). SRR possesses a number of advantages which are very attractive to the implementation of input buffered switch. SRR is on order $O(1)$ which requires minimal computational complexity. Secondly, SRR gives good delay bounds and fairness performance for each session. Thirdly, SRR can decompose a sequence into fixed size groups. In this way, by maintaining a SRR scheduler in each output port, scheduling can be performed in a distributed manner which largely reduces the complexity of the algorithm.

1 Introduction

The development of multirate interconnection networks comes from the necessity of developing a new generation of switches for broadband services which require stringent Quality of Services (QoS) guarantees, such as end-to-end delay, jitter and minimum bandwidth requirements. The nonblocking conditions for multirate traffic with different types of networks and comparisons on their complexity are established in [1]. Related researches [2] [3] [4] [5] [6] [7] have been carried out but these studies do not give us a complete set of traffic scheduling and routing algorithm to guarantee the QoS requirement.

A novel routing scheme for large-scale packet switches called path switching was proposed in [8]. This scheme provides end-to-end QoS guarantees in Clos network. Path switching is a compromise of static routing and dynamic routing schemes. The basic idea of this scheme is to use a set of predetermined connection patterns of central switching modules, and these connection patterns are used repeatedly in a periodical manner. The capacity requirement on each session can be satisfied in the long run, and the computation of route assignment on-the-fly can be avoided.

* The work described in this paper was substantially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region. (Project no. CUHK4380/02E and Direct Grant 2005/06 2050360).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 256–267, 2006.

However, the output traffic of this scheme may become bursty. In path switching, a token is considered as a middle module in a particular time slot through which packets can transverse from a particular input module to a particular output module. In this case, tokens are assigned to each input module to satisfy all their capacity requirement. While this assignment problem can be solved by edge-coloring of bipartite graph, this easy solution cannot guarantee an uniform distribution of tokens, which is necessary to achieving smooth output traffic and tight delay bounds for each session.

Recently, the traffic matrix decomposition approach of path switching [8] was adopted by Chang et al [9] [10], in which a scheduling algorithm for QoS guarantees of input queued switches was developed. Their algorithm consists of two parts: an offline part that breaks down the rate matrix into a set of permutation matrices, and an online part that schedules these matrices using Weighted Fair Queueing (WFQ). However, the time complexity of this algorithm is quite large (the offline part is of $O(N^{4.5})$ and the online part is of $O(logN)$ for a $N \times N$ switch). The number of permutation matrices that results from this decomposition is in the order of N^2. Moreover, the worst case delay can be very large since the decomposition is done randomly and the resulting permutation matrices would not be able to provide smooth output traffic for every session. The load balanced approach in [10], although is much simpler, would give out-of-order packets problem. Therefore, this algorithm is not quite practical for large scale packet switch.

In this paper, a set of scheduling algorithm and admission control strategy is provided in order to guarantee smoother output traffic while maintaining low operational complexity. This set of traffic control algorithm is based on a fair scheduling algorithm called Smoothed Round Robin (SRR) [11]. In Section 2, the basic concept of SRR will be explained. In Section 3, we will explain the methodology of implementing SRR in input-buffered switch and also the admission control strategy needed. In Section 4, the performance of the scheduler will be discussed. By applying network calculus, the deterministic QoS guarantees are derived in Section 5. At last, we will conclude our work in Section 6.

2 The Smoothed Round Robin

Smoothed Round Robin is a simple scheduling algorithm which has the major advantage of its O(1) time computational complexity. Two key data structures of the scheduler are the Weight Spread Sequence (WSS) and the Weight Matrix (WM). The WSSs are defined as follows:

1) The first WSS $S^1 = 1$.

2) The kth WSS is

$$S^k = S^{k-1}, k, S^{k-1} \tag{1}$$

where $k > 1$ and $1 \leq i \leq 2^k - 1$.

For the Weight Matrix, each flow is assigned with a weight in proportion to its reserved rate and the set of weights is assumed to be {1, 2, 3, ..., 2^k -1}. Then the weight of $flow_f$ can be coded in binary as

$$w_f = \sum_{n=0}^{k-1} a_{f,n} 2^n, \text{where } a_{f,n} = \{0, 1\}.$$

The Weight Vector of $flow_f$ is defined as

$$WV_f = \{a_{f,(k-1)}, a_{f,(k-2)}, ..., a_{f,0}\}. \tag{2}$$

Then the Weight Matrix is defined as

$$WM = \begin{bmatrix} WV_1 \\ WV_2 \\ WV_3 \\ \vdots \\ WV_N \end{bmatrix} \tag{3}$$

for N input flows. The columns of the Weight Matrix are named as $column_{k-1}$, $column_{k-2}$, $\ldots$, $column_0$ from left to right respectively. Notice that k is the number of columns in the WM. To schedule packets, SRR scans the WSS sequence term by term. When the value of the term is i, the $column_{k-i}$ of the WM is chosen. In this column, the scheduler will scan the terms from top to bottom. When the term is not 0, the scheduler will serve the corresponding flow.

For example, given there are three flows with weights $w_a = 2$, $w_b = 3$, $w_c = 5$, the Weight Matrix and Weighted Spread Sequence are

$$WM = \begin{bmatrix} WV_a \\ WV_b \\ WV_c \end{bmatrix} = \begin{bmatrix} 0\,1\,0 \\ 0\,1\,1 \\ 1\,0\,1 \end{bmatrix} \text{ and } WSS = 1, 2, 1, 3, 1, 2, 1$$

In this way, the first column would be served first, which only contains flow C. Then the second column get served, which contains flows A and B. The overall result, CABCBCCABC, would be generated when the whole WSS was processed.

3 Scheduling in Input Buffered Switch by Applying the Concept of SRR

The concept of path switching is adopted here, where the time axis is divided into frames of time-slots and tokens (port-to-port path) are assigned to input flows to fulfill their capacity requirements. To provide input buffered switch with smooth output traffic and deterministic QoS guarantees, tokens have to be assigned uniformly. This can be achieved by incorporating SRR into input buffered switch. Notice that SRR is originally designed to support variable size packets by means of deficit counter. It can be easily adopted in the switching environment which requires fixed-sized packets.

3.1 Admission Control

To begin with, a set of traffic admission control has to be set up. A weight matrix is maintained at each input/output port such that the weight of an incoming flow is recorded in the WMs of input port as well as its destined output port. However, the column sum of these WMs are predefined such that an incoming request is rejected if the inclusion of its weight into the WMs would violate the column sum restriction, either at the input or the output port. All the WMs are having the same set of column sum restriction.

3.2 Token Assignment

To assign tokens, SRR would be performed in each output port. However, tokens cannot be assigned simply according to the result of SRR since different output port may reserve token for the same input port but each input port can only process one packet. In this way, permutation matrix cannot be formed in that particular time-slot. This is the reason why token assignment algorithm has to be centralized as it has to cooperate with both the input and output port. However, with the restrictions in the WMs, we can have an easier solution. The restriction in the output port has allowed a simple partitioning of tokens into groups and the restriction in the input port has allowed a simple small-scale rescheduling within each group to form the permutation matrices.

For example, given the following capacity requirement matrix

$$R = \begin{bmatrix} 7\,5\,2\,2 \\ 0\,6\,7\,3 \\ 3\,3\,6\,4 \\ 6\,2\,1\,7 \end{bmatrix},$$

which tells us the weight requirement from each input port to each output port, the Weight Matrices for output ports (columns of R) are

$$\begin{bmatrix} 1\,1\,1 \\ 0\,0\,0 \\ 0\,1\,1 \\ 1\,1\,0 \end{bmatrix} \begin{bmatrix} 1\,0\,1 \\ 1\,1\,0 \\ 0\,1\,1 \\ 0\,1\,0 \end{bmatrix} \begin{bmatrix} 0\,1\,0 \\ 1\,1\,1 \\ 1\,1\,0 \\ 0\,0\,1 \end{bmatrix} \begin{bmatrix} 0\,1\,0 \\ 0\,1\,1 \\ 1\,0\,0 \\ 1\,1\,1 \end{bmatrix}$$

Notice that the column sums of these WMs are identical (2, 3, 2). This satisfies the output port restriction. On the other hand, for the input ports (rows in R), the WMs are

$$\begin{bmatrix} 1\,1\,1 \\ 1\,0\,1 \\ 0\,1\,0 \\ 0\,1\,0 \end{bmatrix} \begin{bmatrix} 0\,0\,0 \\ 1\,1\,0 \\ 1\,1\,1 \\ 0\,1\,1 \end{bmatrix} \begin{bmatrix} 0\,1\,1 \\ 0\,1\,1 \\ 1\,1\,0 \\ 1\,0\,0 \end{bmatrix} \begin{bmatrix} 1\,1\,0 \\ 0\,1\,0 \\ 0\,0\,1 \\ 1\,1\,1 \end{bmatrix}$$

Again, the column sums here are also identical. This satisfies the input port restriction. By performing SRR at each output port of the switch, we have

A D	A C D	A D	A C	A D	A C D	A D
A B	B C D	A B	A C	A B	B C D	A B
B C	A B C	B C	B D	B C	A B C	B C
C D	A B D	C D	B D	C D	A B D	C D

where A, B, C and D represent the tokens for the four input ports. Notice that permutation matrices cannot be formed in each time-slot. However the tokens are partitioned in the same format and by shuffling the tokens inside each partition (for example, by edge coloring of bipartitle graph [8]), we have

A D	A C D	A D	A C	A D	A C D	A D
B A	C D B	B A	C A	B A	C D B	B A
C B	B A C	C B	B D	C B	B A C	C B
D C	D B A	D C	D B	D C	D B A	D C

In this way, the tokens are distributed uniformly and thus output traffic is smoother. Since the number of elements in each group should be small and the rescheduling within group can be done in parallel fashion, complexity of this algorithm should also be small.

4 Performance Analysis

In this section, we would first discuss the relative fairness of the SRR, which is essential in obtaining deterministic QoS guarantees, which will be discussed in the next section. The scheduling delay bound of the proposed scheduler would also be discussed here.

4.1 Relative Fairness of Smoothed Round Robin

To analyze the fairness of scheduling algorithm, Golestani [12] proposed to find the maximum difference between the normalized service received by two backlogged flows over any time interval as a fairness index, which can be expressed as

$$RF = \max\left(\left|\frac{V_f(\tau,t)}{w_f} - \frac{V_g(\tau,t)}{w_g}\right|\right)$$

where $V(\tau,t)$ represents the amount of service received by a session in any time interval τ to t and w as the weight of that session. Given k is the order of the current WSS used by SRR, the author has showed in [11] that

$$\left|\frac{V_f(0,t)}{w_f} - \frac{V_g(0,t)}{w_g}\right| \leq \frac{k}{2\min(w_f,w_g)}, \tag{4}$$

which does not represent the real relative fairness index. This is because in (4), it is assumed that the backlog would always start from the beginning of WSS, which is not true. For example, when $w_f = 3$ and $w_g = 2$, the output is F, G, F, F, G and

$$\frac{V_f(0,t)}{w_f} - \frac{V_g(0,t)}{w_g} = \left\{\frac{1}{3}, -\frac{1}{6}, \frac{1}{6}, \frac{1}{2}, 0\right\}$$

for t={1, 2, 3, 4, 5}. If the backlog start at $t = 2$, the output sequence becomes F, F, G, F, G and then

$$\frac{V_f(2,t)}{w_f} - \frac{V_g(2,t)}{w_g} = \left\{\frac{1}{3}, \frac{2}{3}, \frac{1}{6}, \frac{1}{2}, 0\right\}$$

for $t = \{3, 4, 5, 6, 7\}$. In this case, the value $\frac{2}{3}$ is larger than the bound $\frac{1}{2}$ in (4).

While we cannot obtain the value of RF directly from (4), we are not far from the solution. As shown in Fig. 1, the value of RF is the sum of LRF and SRF, which represent the larger and smaller parts of RF away from the x-axis respectively. Now, as LRF is actually given in (4), we only have to find the value of SRF, which turns out to be having a smaller bound than LRF.

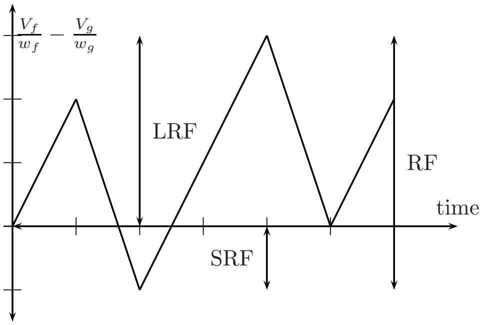


Fig. 1. Relative Fairness - Definition of LRF and SRF

Theorem 1. *For any pair of backlogged flows f and g in SRR, at any time instance t, we have*

$$SRF \leq \frac{(k-1)\max(w_f, w_g) + 2}{2w_f w_g} \tag{5}$$

where k is the order of the current WSS used by SRR.

Proof. The proof is shown in the Appendix. □

Theorem 2. *For any pair of backlogged flows f and g in SRR, where the backlog started at time τ and at any time instance t, we have the relative fairness index*

$$RF = \max\left(\left|\frac{V_f(\tau,t)}{w_f} - \frac{V_g(\tau,t)}{w_g}\right|\right) \leq \frac{(2k-1)\max(w_f, w_g) + 2}{2w_f w_g} \tag{6}$$

where k is the order of the current WSS used by SRR.

Proof.

$$\begin{aligned}RF &\leq LRF + SRF \\ &\leq \frac{k}{2\min(w_f, w_g)} + \frac{(k-1)\max(w_f, w_g) + 2}{2w_f w_g} \\ &= \frac{(2k-1)\max(w_f, w_g) + 2}{2w_f w_g}\end{aligned}$$

Therefore Theorem 2 is proved. □

4.2 Delay Bound of the Proposing Scheduling Algorithm

In this section, we will show the single packet delay bound of this scheduler, which represents the time for a head of line packet to be completely transmitted over the switch. This bound is valid without any input traffic constraint, such as traffic envelope.

Theorem 3. *Suppose the weight assigned to $flow_f$ is w_f, the delay encountered by this flow in the input buffered switch using SRR is bounded by*

$$d_{SRR} \leq \frac{2(w_f + w_G)}{w_f} + 2N_b \tag{7}$$

where w_G is the weight of flows other than $flow_f$ and N_b is the maximum column sum in WM.

Proof. A flow is visited when one of its coefficients is visited by SRR. Therefore, the delay bound of a flow is the maximum value of the intervals between two adjacent visits by SRR. By assuming $2^i \leq w_f \leq 2^{i+1} - 1$, the chain with the maximum length between two adjacent occurrences of element $(k-i)$ is $S^{k-i-1}, (k-y), S^{k-i-1}$ for $y < i$ and $a_{f,y} = 0$. In [11], the author has shown that the delay (V_{cnt}) as mapped by $S^{k-i-1}, (k-y), S^{k-i-1}$ and $column_i$ is

$$V_{cnt} \leq \frac{1}{2^i}\left(w_f + w_G - \sum_{n=0}^{i-1} 2^n \sum_{m=1}^{N} a_{m,n}\right) + \sum_{m=1}^{N} a_{m,y} \tag{8}$$

When applied in a switch, the extra delay resulted is the number of elements in $column_i$, thus

$$d_{SRR} \leq \frac{1}{2^i}\left(w_f + w_G - \sum_{n=0}^{i-1} 2^n \sum_{m=1}^{N} a_{m,n}\right) + \sum_{m=1}^{N} a_{m,y} + \sum_{m=1}^{N} a_{m,i} \tag{9}$$

$$\leq \frac{1}{2^i}(w_f + w_G) + \sum_{m=1}^{N} a_{m,y} + \sum_{m=1}^{N} a_{m,i} \tag{10}$$

$$\leq \frac{2(w_f + w_G)}{w_f} + N_b + N_b \tag{11}$$

Hence, Theorem 3 is proved. □

5 Deterministic QoS Guarantees

A model to study the deterministic QoS guarantees for each session is developed. Using this model, upper bounds on delay and backlog can be established, assuming each input traffic stream is under leaky-bucket rate control and there is no packet loss due to buffer overflow and delay bound violation.

5.1 Network Calculus

To model the service received by each session, we can use the concept of service curve [13], which represents the least amount of service provided by the network element to a data session during its busy period. Fig. 2 shows an arrival curve $A()$ together with a service curve $S()$. The service curve is a straight line with the slope representing the service rate reserved for a particular session at the network element. In this way, the delay and backlog bounds can be easily calculated as shown in the figure.

As discussed in our previous work [14], given an arrival curve with burstiness constraint (σ, ρ, C), where σ is the bucket size, ρ is the arrival rate and C is the maximum rate of tokens flowing out from the bucket, the delay of a packet is bounded by

$$D = v + \frac{\sigma}{C-\rho}\left(\frac{C}{g-1}\right) \tag{12}$$

while the backlog bound is

$$B = \begin{cases} vg + \frac{C-g}{C-\rho}\sigma & \text{if } v < \frac{\sigma}{C-\rho} \text{ and } C > g \\ vC & \text{if } v < \frac{\sigma}{C-\rho} \text{ and } C \le g \\ v\rho + \sigma & \text{otherwise} \end{cases} \tag{13}$$

where g denotes the reserved rate of the session and v is a parameter related to the service curve as shown in Fig. 2 and will be discussed in the next subsection.

5.2 Obtaining the Service Curve of the Proposing Scheduling Algorithm

As shown in Fig. 3, by drawing all the possible schedulers output on the same graph, a service curve, which is the lower bound of all the distributions can be obtained. Denotes w_f as the weight of the session we are interested in and w_G as the aggregated weights of other sessions sharing the same output port. Denotes $V_f(\tau, t)$ as the packets served for the session of interest from the start of backlog τ to time t and $V_G(\tau, t)$ as the packets served for other sessions, then the value of v, which denotes the minimum delay needed to guarantee a steady service of reserved rate from the switch, can be obtained. According to Fig. 3, we have

$$\frac{V_f(\tau, t)}{V_f(\tau, t) + V_G(\tau, t) - v} \ge \frac{w_f}{w_f + w_G} \tag{14}$$

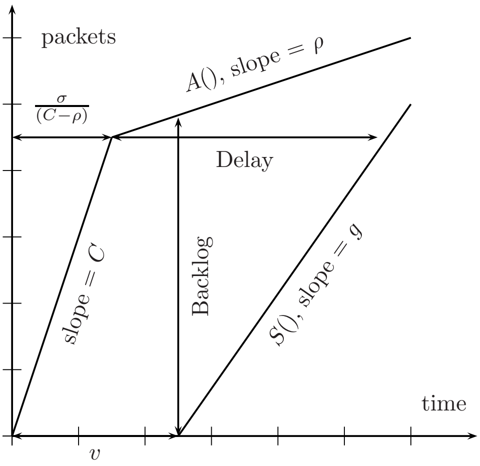


Fig. 2. Arrival and Service Curves

which implies that

$$v \geq \frac{V_G(\tau, t)w_f - V_f(\tau, t)w_G}{w_f} \tag{15}$$

Therefore, the smallest value of v can be obtained once we can find the maximum value of $V_G(\tau, t)w_f - V_f(\tau, t)w_G$.

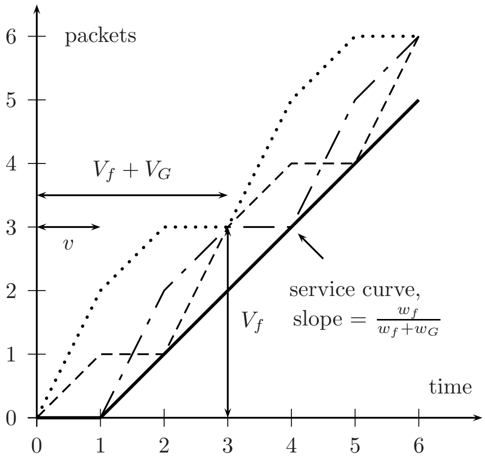


Fig. 3. Resultant Service Curve

Theorem 4. *Given w_f as the weight of the session we are interested in and w_{g_i} as the weights of other flows destined to the same output port as $flow_f$, v is bounded by*

$$v \geq \sum_i \frac{(2k_i + 1)\max(w_f, w_{g_i}) + 2}{2w_f}$$

where k_i are the smallest integers that $2^{k_i} \geq \max(w_f, w_{g_i})$.

Proof.

$$\begin{aligned} |V_G w_f - V_f w_G| &\leq \sum_i [|V_{g_i} w_f - V_f w_{g_i}| + \max(w_f, w_{g_i})] && (16)\\ &\leq \sum_i \left[\frac{(2k_i - 1)\max(w_f, w_{g_i}) + 2}{2} + \max(w_f, w_{g_i})\right] && (17)\\ &\leq \sum_i \frac{(2k_i + 1)\max(w_f, w_{g_i}) + 2}{2} && (18) \end{aligned}$$

The last term of (16) represents the extra delay resulted from the rescheduling within each group. (6) was applied to give (17). Then from (15), we have

$$\begin{aligned} v &\geq \frac{V_G(\tau, t)w_f - V_f(\tau, t)w_G}{w_f} && (19)\\ &\geq \sum_i \frac{(2k_i + 1)\max(w_f, w_{g_i}) + 2}{2w_f} && (20) \end{aligned}$$

Thus, Theorem 4 is proved. □

6 Conclusion

In this paper, we applied smoothed round robin to input buffered switches and yielded desired result. SRR has short delay bounds and good fairness performance, thus the application of SRR in input buffered switch will allow a less-bursty output traffic. We have derived the relative fairness of SRR and the deterministic QoS guarantees of the proposed scheduling algorithm by using the concept of Network Calculus.

References

1. Melen, R., Turner, J.S.: Nonblocking networks for fast packet switching. In: IEEE Infocom '89. Volume 2. (1989) 548–557
2. Li, S., Ansari, N.: Input-queued switching with QoS guarantees. In: IEEE Infocom'99. (1999) 1152–1159
3. Hung, A., Kesidis, G., Mckeown, N.: ATM input-buffered switches with guaranteed-rate property. In: IEEE ISCC'98. (1998) 331–335
4. Liotopoulos, F., Chalasani, S.: Semi-rearrangeably nonblocking operation of Clos networks in the multirate environment. IEEE Transactions on Networking. **4** (1996) 281–291
5. Naraghi-Pour, M., Hegde, M., Suresh, S.: Scheduling multi-rate traffic in a time-multiplex switch. In: Proceedings on Information Theory'94. (1994) 406

6. Valdimarsson, E.: Blocking in multirate interconnection networks. IEEE Transactions on Networking. **42** (1994) 2028–2035
7. Favalli, L.: Rearrangeability conditions for multirate Benes networks. In: GLOBECOM' 93. (1993) 734–738
8. Lee, T., Lam, C.: Path switching: A quasi-static routing scheme for large-scale ATM packet switches. IEEE Journal on Selected Areas in Communications **15**(5) (1997) 914–924
9. Chang, C.S., Chen, W.J., Huang, H.Y.: Birkhoff-von Neumann input-buffered crossbar switches for guaranteed-rate services. IEEE Trans. Commun. **49** (2001) 1145–1147
10. Chang, C.S., Chen, W.J., Huang, H.Y.: Providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches. In: IEEE Infocom 03. Volume 3. (2003) 1622–1632
11. Guo, C.: SRR: An O(1) time-complexity packet scheduler for flows in multiservice packet networks. IEEE Transactions on Networking. **12**(6) (2004) 1144–1155
12. Golestani, S.: A self-clocked fair queueing scheme for broadband applications. In: IEEE Infocom 94. (1994) 636–646
13. Cruz, R.L.: Quality of service guarantees in virtual circuit switched networks. IEEE Journal on Selected Areas in Communications **13** (1995) 1048–1056
14. Chan, M.C., To, P., Lee, T.T.: Per-connection performance guarantees for cross-path ATM packet switch. In: ATM Workshop 1999. (1999) 469–474

Appendix. Proof of Theorem 1

For easier presentation, we multiply both sides of (5) with $w_f w_g$, then we define

$$WSRF = (w_f w_g)SRF \leq \frac{k-1}{2}\max(w_f, w_g) + 1$$

This theorem is proved by induction as follows

1) It is true for $k = 1$ and 2,

2) Suppose that the inequality is correct using a kth WSS, i.e., for any pair of w_f and w_g, we have

$$WSRF \leq \frac{k-1}{2}\max(w_f, w_g) + 1$$

Then for any pairs of f' and g' using a $(k+1)$th WSS, $w_{f'}$ and $w_{g'}$ can be expressed as

$$\begin{aligned} &w_{f'} = 2w_f + a_{f',0}, w_{g'} = 2w_g + a_{g',0} \\ &\qquad \text{where } w_{f'} > 1, w_{g'} > 1, a_{f',0}, a_{g',0} = 1 \text{ or } 0. \end{aligned}$$

Therefore, the service sequence of flow f' and g' can be expressed as

$$S^{k+1}(f', g') = S^k(f, g), \{a_{f',0}.f, a_{g',0}.g\}, S^k(f, g).$$

Here we just show the last case where $w_{f'} = 2w_f{+}1$ and $w_{g'} = 2w_g\ {+}1$.

When $V_{f'} = V_f$ and $V_{g'} = V_g$,

$$\begin{aligned}
WSRF &= |(2w_f + 1)V_g - (2w_g + 1)V_f| \\
&\leq |2w_f V_g - 2w_g V_f| + |V_g - V_f| \\
&\leq [(k-1)\max(w_f, w_g) + 2] + [\max(w_f, w_g)] \\
&\leq k \max(w_f, w_g) + 2 \\
&\leq \frac{k}{2}\max(2w_f, 2w_g) + \frac{k}{2} + 1 && \text{(for } k \geq 2\text{)} \\
&\leq \frac{k}{2}\max(w_{f'}, w_{g'}) + 1 && (w_{f'} = 2w_f + 1)
\end{aligned}$$

When $V_{f'} = w_f + 1$ and $V_{g'} = w_g$,

$$\begin{aligned}
WSRF &= |(2w_f + 1)w_g - (2w_g + 1)(w_f + 1)| \\
&= |w_g + w_f + 1| \\
&\leq 2\max(w_f, w_g) + 1 \\
&\leq \frac{k}{2}\max(w_{f'}, w_{g'}) + 1 && \text{(for } k \geq 2\text{)}
\end{aligned}$$

When $V_{f'} = w_f + 1$ and $V_{g'} = w_g + 1$,

$$\begin{aligned}
WSRF &= |(2w_f + 1)(w_g + 1) - (2w_g + 1)(w_f + 1)| \\
&= |w_g - w_f| \\
&\leq \frac{k}{2}\max(w_{f'}, w_{g'}) + 1
\end{aligned}$$

When $V_{f'} = w_f + 1 + V_f$ and $V_{g'} = w_g + 1 + V_g$,

$$\begin{aligned}
WSRF &= |(2w_f + 1)(w_g + 1 + V_g) - (2w_g + 1)(w_f + 1 + V_f)| \\
&\leq |2w_g V_f - 2w_f V_g| + |w_g - V_g - w_f + V_f| \\
&\leq [(k-1)\max(w_f, w_g) + 2] + [\max(w_f, w_g)] && \text{(since } w_f \geq V_f\text{)} \\
&\leq k\max(w_f, w_g) + 2 \\
&\leq \frac{k}{2}\max(w_{f'}, w_{g'}) + 1
\end{aligned}$$

Hence, for different combinations of $V_{f'}$ and $V_{g'}$ in the last case, we have

$$\begin{aligned}
WSRF &\leq \frac{k}{2}\max(w_{f'}, w_{g'}) + 1 \\
SRF &\leq \frac{k\max(w_{f'}, w_{g'}) + 2}{2w_{f'}w_{g'}}
\end{aligned}$$

Therefore, Theorem 1 follows by induction. □

VoD QAM Resource Allocation Algorithms

Jiong Gong[1], David Reed[1], Terry Shaw[1], Daniel Vivanco[1], and Jim Martin[2]

[1] Cable Television Laboratories, Inc.,
858 Coal Creek Circle, Louisville, CO 80027
j.gong@cablelabs.com, d.reed@cablelabs.com,
t.shaw@cablelabs.com, d.vivanco@cablelabs.com
[2] Department of Computer Science,
Clemson University, USA
jim.martin@cs.clemson.edu

Abstract. This paper proposes a new Quadrature Amplitude Modulation (QAM) resource allocation algorithm for Video on Demand (VoD) when there is a mixture of standard definition (SD) and high definition (HD) video streams. We have developed a simulation model to compare this algorithm with two popular algorithms: the least-loaded algorithm and the most-loaded algorithm. We show that our algorithm, which we call the non-mixing algorithm, performs significantly better than the two existing algorithms by accommodating more streams thereby lowering the blocking probabilities under a range of assum-ptions of peak concurrent usage rate and percentage of HD streams. Using computer simulation we found that the non-mixing algorithm leads to an average of 4.39% higher allowed peak usage rate than the least-loaded and most-loaded algorithms.

Keywords: VoD, HFC networks, Broadband access, Capacity planning, Congestion control, Traffic management & control, Traffic modeling & characterization, Resource allocation, Network modeling & simulation.

1 Introduction

Video on Demand (VoD) systems over broadband access networks are likely to see a significant change in usage patterns over the next few years. The percentage of high definition (HD) VoD stream requests is likely to increase significantly from zero to approximately 10%, and peak usage is likely to increase significantly from the current average of 5% to approximately 30% as subscription-based VoD (SVoD) and digital video recorder (DVR) applications become more mainstream[1] [6]. Cable operators will require a detailed understanding of the impact of these changes in the provisioning process.

When a cable subscriber purchases a VoD selection, the video stream is assigned to a QAM modulator over a specified 6 MHz RF channel. The encoding rate of the stream along with the specific QAM configuration determines the aggregate number of streams that can be assigned to the channel. For example over a 256 QAM modulated

[1] From a commercial North American Cable Operator's market forecast. One cable operator is lately seeing close to 10% peak usage rate after the introduction of sVoD service.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 268 – 280, 2006.

channel, if all content is in SD format (i.e., MPEG2) and is encoded at a constant bit rate of 3.75 Mbps, 10 streams can be assigned to the same channel and thus all of the channel bandwidth is used. A VoD system, referred to as a **service group**, consists of content servers, a delivery network, a number of QAM modulators and a set of subscribers. During the purchase of a VoD selection, the resource allocation algorithm must assign a new stream to one of the modulators in the service group. If the channel capacity is an integral multiple of the bandwidth consumed by an SD flow, the QAM resource allocation algorithm is trivial. However in future VoD systems, there will be a mix of standard definition (SD) streams and HD streams.

A common encoding rate for HD streams is 12.5 Mbps. Assuming a channel capacity of 37.5 Mbps based on 256 QAM modulation, three HD streams would completely fill a channel. The difficulty comes when a combination of SD and HD streams are assigned to a channel. In this case, some amount of the channel bandwidth will be unused. The worst case percentage of stranded bandwidth (B_S) is $B_S = \left(\frac{r_h - r_s}{Q} \right)$, where r_s and r_h denote the streaming bit rate for SD and HD streams respectively, and Q is the channel capacity [2]. In the worst case, each QAM modulator has just under r_h bandwidth stranded. This could occur if a series of HD stream requests arrive that almost fills the QAM (i.e., to the point where one more HD request would completely fill the QAM), but then an SD stream request arrives and gets allocated. For the 256 QAM scenario described above, up to 23.3% of the channel bandwidth could be stranded.

The current prevailing QAM allocation methods include two algorithms; one that allocates incoming streams starting from the lightest-loaded QAM modulator and one that starts from the busiest-loaded QAM modulator. In the rest of the paper, we refer to the former as the "least-loaded" algorithm and the latter as the "most-loaded" algorithm. It is generally believed that the most-loaded algorithm performs better than the least-loaded algorithm when there is the presence of HD VoD streams, a fact that is confirmed in our analysis. We propose and evaluate a new QAM resource scheduling algorithm called the "non-mixing" algorithm. In this paper, we present the results of a simulation-based analysis that suggest that the non-mixing algorithm can allow peak usage rates 4.39% higher than most-loaded algorithm. A further contribution of this paper is the results of a VoD usage modeling effort which was necessary to exercise our simulation model in a realistic manner.

This paper is organized as follows. Related work is presented in section 2. The VoD usage model and the proposed non-mixing algorithm are presented on section 3 and 4, respectively. In sections 5 and 6 we present our analysis methodology and simulation-based results, respectively. Finally section 7 presents the conclusion of the analysis and identifies future work items.

2 Related Work

A large amount of prior research has addressed the scalability of large-scale VoD systems. Techniques have been identified that reduce the resources that are required persession. Batching requires users to wait in a group for the same content for a predetermined amount of time and then serves them in a batch using a single multicast channel [4] [5] [1]. Periodic broadcasting schedules the transmission of content over multiple channels in periodic intervals allowing arriving users to join the next cycle [3] [12] [7]. Patching attempts to merge users who are on separate channels to an existing multicast channel [11] [13]. Piggybacking merges users on separate channels by slightly changing playback rates of users in an effort to have everyone get to the same point in the stream at which time the separate channels would be exchanged for a single multicast channel [8] [14]. While these ideas are likely to be relevant in future cable VoD systems, most current deployments are relatively small in scale. Provisioning the optimal number of QAM modulators in a VoD service set is generally based on the rule of thumb that says about 5% of the total subscriber population will use VoD during peak periods. There has been industry discussion on QAM allocation algorithms [10] [9]. However, to the best of our knowledge, there has not been an academic evaluation of QAM resource allocation algorithms.

The QAM allocation problem is essentially a **bin packing** problem. The classic bin packing algorithm packs a list of items $L = (a_1, a_2, ..., a_n), a_i \in (0,1]$ for all i, into the minimum number of bins each with a capacity of 1. The least loaded QAM allocation algorithm is a form of best fit packing and the most loaded allocation algorithm is a form of worst fit packing [2]. In brief, a best fit packing algorithm selects the bin that has the most free space and the worst fit algorithm selects the bin that has the least free space. The standard metric that is used to evaluate bin packing algorithms is a measure of the number of bins that are required to pack various input lists. The ratio of the number of bins required by the algorithm under study to the number of bins required by an optimal algorithm (i.e., an off line algorithm) is known as the *R* value. It has been shown that both the best fit and worst fit algorithms have an *R* value of 2 [2]. In the QAM allocation problem domain, the number of bins is fixed. Items in bins may leave after an amount of time (i.e., when the subscriber finishes watching the movie the stream is removed from the QAM). Rather than use the *R* metric, we are interested in the probability that a stream's request is denied due to insufficient capacity. We use the blocking rate to characterize allocation algorithm performance.

3 VoD Model

We have developed a model of VoD usage based on empirical data. The data used in this study were collected from 200 service groups of a large cable operator in North America. The average size of each service group is approximately 500 set-top boxes. Figure 1 illustrates diurnal average usage patterns over the course of one week for all requests. The results show higher usage rate values for Thursday, Friday and Saturday evening from 10:00pm until midnight. Note that the maximum 2% VoD usage rate shown in Figure 1 was the average over all 200 service groups analyzed, while some service groups exhibited peak usage rates close to 5%.

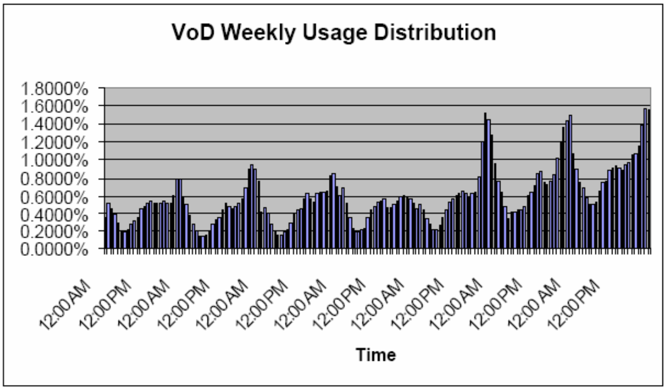


Fig. 1. Weekly VoD Usage (Sunday 12:00am through Sat 12:00am)

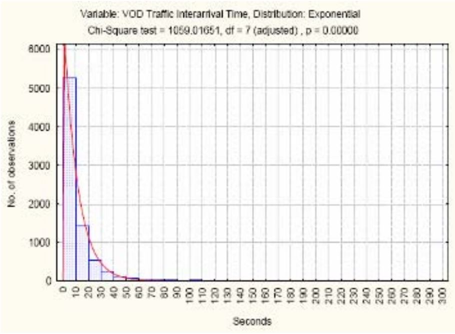


Fig. 2. Histogram and Fitted Exponential Distribution of VoD Request Interarrival Times

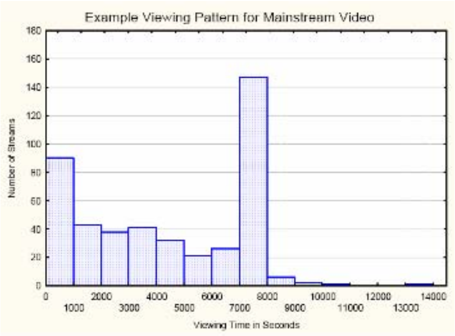


Fig. 3. Distribution of Stream Length for Mainstream Video Titles

We modeled the interarrival times of VoD request streams and their duration. Our results indicate that interarrival times follow an exponential distribution, although each of the main genres of content, including mainstream movies, adult content and video browsing have different fittings. Video browsing refers to short-lived streams mainly generated by a sVoD user who browses the available VoD channels available in his/her subscription package. Figure 2 shows the fit of interarrival VoD request times associated with 7800 instances of stream arrivals observed in the data set. The x-axis represents 10-second windows over a period of 24 hours. The solid line is a fitted exponential distribution curve with a λ of 0.091.

We have also modeled the viewing time distribution of mainstream movies. Our results suggest a mixed probability distribution. As illustrated in Figure 3, there was a significant mode at a viewing time of 2 hours which is the average length of mainstream movies. The data suggests a large number of early exits, which can be associated with video browsing generated by sVoD users.

4 Non-mixing Algorithm

In this section we describe a new QAM allocation algorithm called the non-mixing algorithm. We start by describing a mathematical framework to model the problem. Suppose a collection of n QAM modulators is deployed to serve a VoD service group. Let q_i, $i = 1, 2, ..., n$, denote the used capacity of each QAM modulator i. Total capacity, Q, which is usually 37.5 Mbps for a 256 QAM, is assumed to be the same for all QAM modulators. Therefore, the remaining capacity that can be used for new stream requests on that QAM modulator is then $Q - q_i$. Let r_s and r_h denote the streaming bit rate, respectively, for SD and HD streams. The two types of streams may arrive at a collection of QAM resources according to two distinct random processes, such as the Poisson process, but exit the system based on the same holding time distribution. We call the current state of any given QAM (q_i) at a particular time as an allocation. We define an allocation as inefficient, if,

$$Q - q_i < r_h, \ \forall i, \text{ and } \sum Q - q_i \geq r_h \tag{1}$$

In other words, none of the QAM modulators individually has the capacity, even though the sum of all available resources on each QAM modulator is able to support one or more HD stream requests. A better scheduling algorithm would generate fewer cases of inefficient allocations. Note that while each type of stream is assumed to be in itself modulus in its own bit rate, they jointly are not when they are mixed together in a QAM modulator. As a result, inefficiency tends to arise when different stream types are mixed together. Both most-loaded and least-loaded algorithms lead to mixed allocations at the QAM modulators. In the following lines the non-mixing algorithm is going to be presented. Let's first start by defining 4 possible states for any QAM on the system at any given time, depending on its current allocation;

- No streams have been allocated.
- A mixture of SD and HD streams are occupying it.
- Only SD streams are occupying it.
- Only HD streams are occupying it.

Mathematically, we denote these four types accordingly by defining a state function as:

$$S_i(q_i) = \begin{cases} 1, \ if \, q_i = 0 \\ 2, \ if \, q_i = x_i r_s + y_i r_h, \ x_i \neq 0, \ y_i \neq 0 \\ 3, \ if \, q_i = x_i r_s, \ x_i \neq 0 \\ 4, \ if \ q_i = y_i r_h, \ y_i \neq 0 \end{cases} \tag{2}$$

where x_i and y_i are positive integers representing the number of SD and HD streams, respectively, occupying QAM modulator i. In the above four states, we call a QAM modulator in state 1 an empty QAM modulator. We call a QAM modulator in state 2, that is $S_i(q_i) = 2$, a mixing QAM modulator. QAM modulators in state 3 and

4 are called non-mixing SD and HD QAM modulators, respectively. The algorithm selects a QAM using the following prioritized rules:

- Select a non-mixing QAM modulator of the same stream type.
- Select an empty QAM modulator.
- Select a mixing QAM modulator.
- The last resort is to create another mixing QAM modulator by selecting an existing QAM that currently has only SD or only HD streams.

If there are multiple QAM modulators available within the same state class, priority is given to those QAM modulators that have a larger likelihood of becoming a non-mixing QAM modulator or an empty QAM modulator once some streams start to drop. This implies the following rules:

- If multiple non-mixing QAM modulators are available to a stream request of the same stream type, priority should be given to the busiest non-mixing QAM modulator because other mixing QAM modulators have a higher likelihood of being non-mixing or empty.
- If multiple mixing QAM modulators are available to a SD or HD stream request, priority is given to the busiest mixing QAM modulator, because other mixing QAM modulators have a higher likelihood of being non-mixing or empty.
- If multiple non-mixing QAM modulators are available to a stream request of a different type, that is if a stream request will have to create a new mixing QAM modulator, priority is given to the least busy QAM modulator, because it has the highest likelihood of becoming non-mixing again.

Refer to Appendix A for further details of the algorithm.

5 Analysis Methodology

5.1 Simulation Model

We developed a simulation model with which we can evaluate the performance of a set of QAM allocation algorithms and also be used as a capacity planning tool for cable operators. The model simulates a pool of 256 QAM modulators in a VoD service group. Session requests are either SD or HD streams. SD and HD stream requests have been modeled as independent Poisson processes with interarrival times exponentially distributed. The aggregate stream request is the combination of the SD and HD streams, which also follows a Poisson process with interarrival times exponentially distributed [14]. Equation 3 shows the relationship used to calculate the aggregate VoD request interarrival rate, λ, based on the number of users in a service group and the aggregated concurrent usage rate during the peak hour.

$$\lambda = (Number\_user) * (Peak\_usage\_rate / 3600) \qquad (3)$$

Since the peak-usage rate is defined as the maximum number of stream requests during the peak one hour time period, this parameter was converted from hours into seconds. Equations 4 and 5 represent the SD and HD mean interarrival rates, respectively.

$$\lambda_{SD} = (Percentage\_SD\_streams) * \lambda \tag{4}$$

$$\lambda_{HD} = (Percentage\_HD\_streams) * \lambda \tag{5}$$

Arrival requests have been already classified in section 3 in three genres; mainstream movies, adult content and video browsing, and each of them is characterized by their own unique average duration time. Stream durations for each of these genres have been modeled as independent random variables distributed exponentially. This conclusion has been found from the empirical data shown in Figure 3. Note that this figure shows the aggregate stream duration distribution, thus this is the aggregation of three exponential distributions with different average duration times. Equation 6 shows the aggregate stream duration, μ , based on the weighted average based on the proportions of the genres that make up the streams, where *m* represents the number of stream types (in this case *m*=3).

$$\mu = \sum_{j=1}^{m} (Percentage\_movie\_type_j) * (Average\_duration\_movie\_type_j) \tag{6}$$

The proposed VoD simulation engine presented in this paper replicates a real-word stream processing experience as follows;

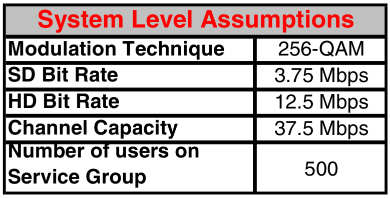
- Accepted streams are released from the QAM modulator when their duration expires.
- Incoming stream requests are compared with the available QAM capacity.
  - If the available capacity is insufficient to handle the request, it is denied and the number of sessions rejected count is incremented by one.
  - If the request is accepted it is placed in an empty channel of one of the available QAM modulators. The channel selection is determined by the stream allocation algorithm that has been configured.

Three allocation algorithms are implemented in the simulation model: least-loaded, most-loaded, and non-mixing. In the most-loaded algorithm, the available QAM capacity remaining within a service group is placed in an array and sorted from the lowest to the highest. The QAM modulator that has the smallest remaining capacity represents the most-loaded or busiest QAM modulator. The incoming stream request is assigned to the most-loaded QAM modulator with sufficient capacity to handle it. In the least-loaded algorithm the reverse occurs, arriving requests are assigned to the QAM modulator that has the largest remaining capacity enough to handle the request. In the non-mixing algorithm, the available QAM capacities are grouped in virtual clusters. The incoming stream is assigned to a QAM channel according to the rules described in section 4.

5.1 Assumptions

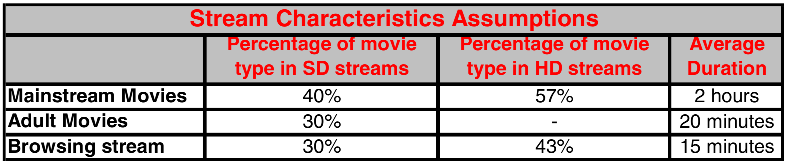
The model we developed can accommodate a great number of scenarios depending on the streaming bit rates, the size of the service group, the precise mixture of SD and

Table 1. System Level Assumptions



System Level Assumptions	
Modulation Technique	256-QAM
SD Bit Rate	3.75 Mbps
HD Bit Rate	12.5 Mbps
Channel Capacity	37.5 Mbps
Number of users on Service Group	500

Table 2. Stream Characteristics Assumptions



Stream Characteristics Assumptions			
	Percentage of movie type in SD streams	Percentage of movie type in HD streams	Average Duration
Mainstream Movies	40%	57%	2 hours
Adult Movies	30%	-	20 minutes
Browsing stream	30%	43%	15 minutes

HD streams and other factors. Table 1 and 2 show the system level assumptions and the stream characteristic assumptions, respectively. The values presented in these tables were obtained from current deployments and real usage VoD patterns data.

6 Results

The performance of the stream allocation algorithms was measured by calculating the average blocking probability first. The analysis varies the peak usage rate, SD and HD stream composition percentages and the QAM pool size in a service group.

Figures 4.a, 4.b and 4.c show the blocking probability for the three mentioned algorithms against a range of peak-usage rates for systems with 4, 8 and 12 QAM modulators, respectively, for the case where the traffic consists of 90% SD streams and 10% HD streams. Figures 5.a, 5.b and 5.c show similar results for the case where the traffic consists of 70% SD streams and 30% HD streams. From these results, it can be seen that non-mixing allocation algorithm leads to a lower blocking probability than the other two algorithms at all usage levels. Filling a QAM modulator with only one type of stream can guarantee maximum capacity utilization given the modular nature of the streaming bit rates. On the other hand, a mixing QAM modulator is likely to have stranded bandwidth that is not sufficient to accommodate an incoming HD stream. Figures 4 and 5 also indicate the poor ability of the least-loaded algorithm to efficiently allocate streams on congested VoD systems. Figure 4.a illustrates that in a VoD system consisting of 10% HD content with 4 QAM modulators, and under 6% peak usage level, the most-loaded and the non-mixing algorithm lead to a blocking probability close to 0%, while the least-loaded algorithm results in a blocking probability close to 4%.

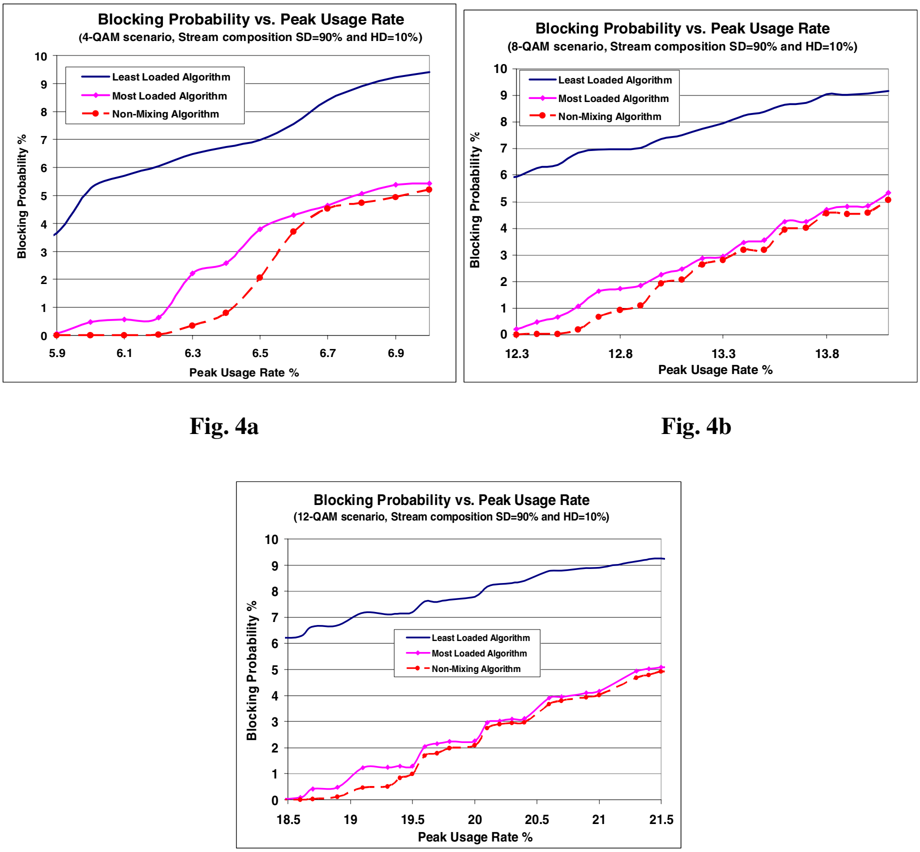


Fig. 4a

Fig. 4b

Fig. 4c

Fig. 4. Blocking Probability vs. Peak-Usage Rate for Least-Loaded, Most-Loaded and Non-Mixing QAM Allocation Algorithms for 90% SD and 10% HD Streams, (a) 4 QAM scenario, (b) 8 QAM scenario, (c) 12 QAM scenario

Figure 6 shows the maximum peak-usage rate that can be supported to meet a 0.3% blocking probability objective in a 4 QAM and 8 QAM VoD systems as a function of the percentage of HD streams. The percentage of capacity improvement of the non-mixing algorithm over the most-loaded algorithm ranges between 3.66% to 5% for the 4 QAM scenario, and between 2.22% to 3.45% for the 8 QAM scenario. For the 4 QAM and 8 QAM scenario an average of 4.39% and 2.71%, respectively, higher allowed peak usage rate can be perceived. As the traffic load increases, it becomes more difficult for the non-mixing algorithm to keep QAM modulators non-mixed. In this case, the non-mixing algorithm has a tendency to behave like the most-loaded algorithm.

Figures 7.a and 7.b show that the maximum peak-usage for 12 and 16 QAM systems respectively when subject to blocking rate objectives of 0.3% and 1%. Most providers would consider a blocking rate of 0.3% acceptable and a 1% rate marginally acceptable. These figures suggest that the maximum peak-usage rate that can be supported to achieve blocking probability objectives decays as the percentage of HD streams increase. To demonstrate how the results from Figure 7 might be used for provisioning, assume that a hypothetical VoD system will experience a peak-usage rate of 20%. For this load, none of the 4 QAM or 8 QAM systems for the 500 home service group can support this volume of traffic, regardless of the stream composition (see Figures 4 and 5). Figure 7.a suggests that a 12 QAM system could handle this load as long as the traffic mix contains less than 7.5% HD streams. Figure 7.b suggests that a 16 QAM system could handle this load for traffic that includes up to 27% HD streams.

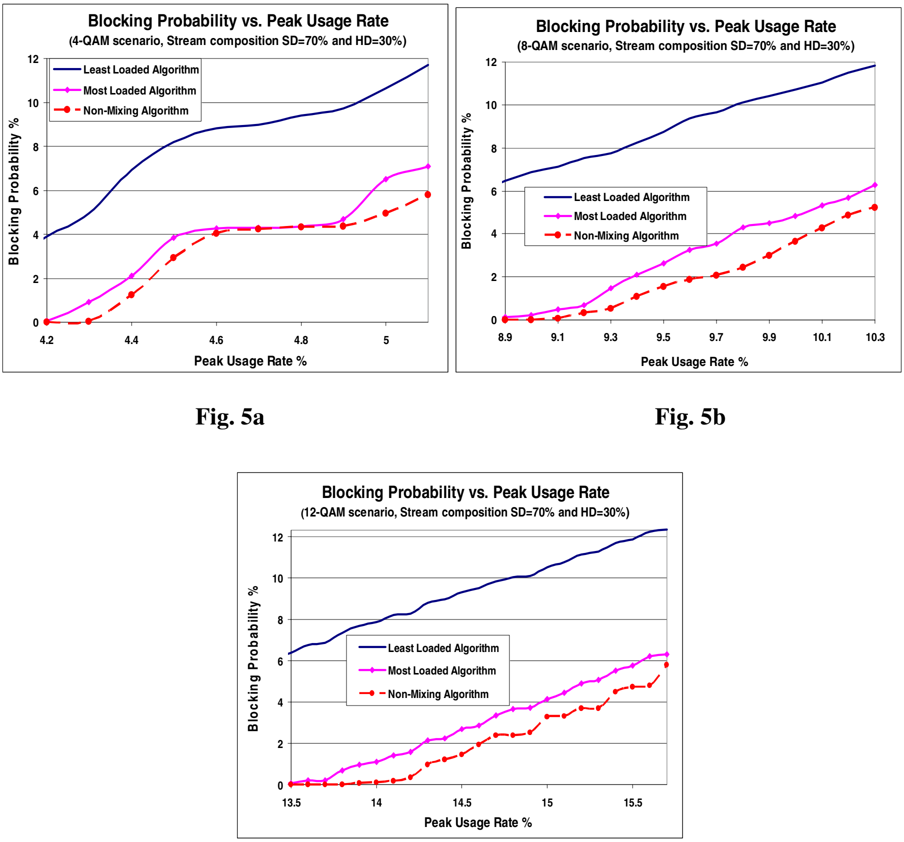


Fig. 5a

Fig. 5b

Fig. 5c

Fig. 5. Blocking Probability vs. Peak-Usage Rate for Least-Loaded, Most-Loaded and Non-Mixing QAM Allocation Algorithms for 70% SD and 30% HD Streams, (a) 4 QAM scenario, (b) 8 QAM scenario, (c) 12 QAM scenario

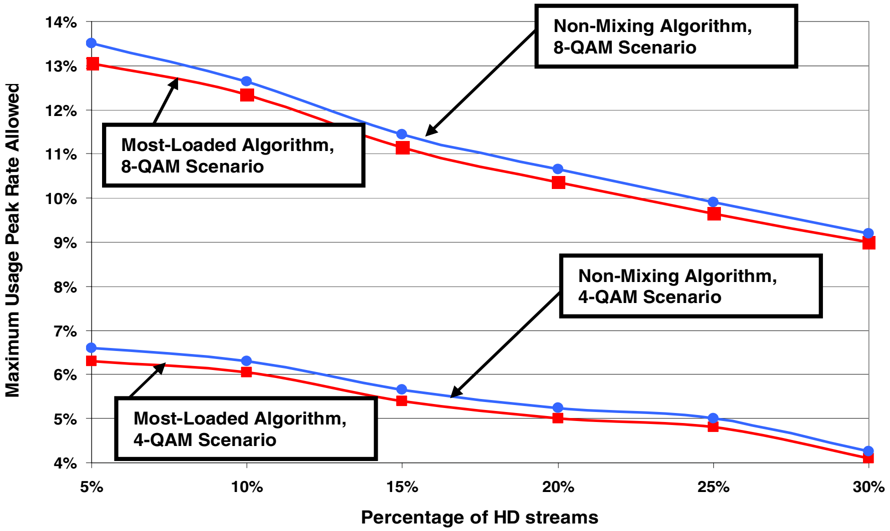


Fig. 6. Maximum Peak-Usage Rate Allowed vs. Percentage of HD streams using No-Mixing and Most-Loaded Algorithm for 4 and 8-QAM Scenarios

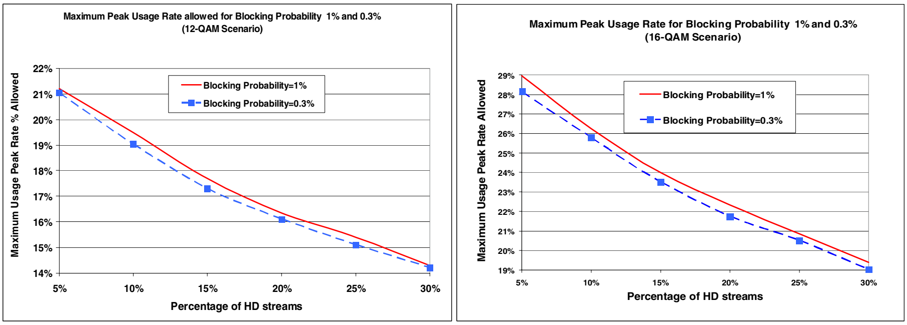


Fig. 7a **Fig. 7b**

Fig. 7. Maximum Peak-Usage Rate Allowed vs. Percentage of HD streams using No-Mixing Algorithm, (a) 12-QAM Scenario, (b) 16-QAM Scenario

7 Conclusions

Our results highlight the effect that QAM allocation algorithms can have on the efficiency of a VoD system. The two commonly deployed algorithms, least-loaded and most-loaded, are designed for current generation VoD systems that offer only SD streams. Future systems will involve a mix of SD and HD streams. We have shown that a least-loaded algorithm can result in more than a five fold increase in blocking probability compared to a most-loaded algorithm when subject to varying levels of SD and HD stream requests. Our proposed algorithm, the non-mixing algorithm, is able to demonstrate better performance under all cases of usage level and under all cases of HD percentage assumptions.

Many VoD systems are deployed using 4 QAM modulators in a service group. Our analysis shows that changing to the non-mixing algorithm can support up to 6.2% peak-hour concurrent usage that contains 10% HD streams, which is difficult to be accommodated with most-loaded or least-loaded algorithms (see figure 4.a). With more HD content, the non-mixing algorithm can generate an average of 4.39% higher allowed peak usage rate over most-loaded algorithm. 6.2% seems to be a reasonable peak-hour concurrent usage assumption in the near term for many VoD systems in North America that are currently experiencing peak-hour concurrent usage below 5%.

The benefits of the non-mixing algorithm over the most-loaded algorithm depend primarily on its ability to avoid, to the extent possible, mixing SD and HD streams. This is driven by several factors, including SD and HD traffic composition, SD and HD streaming bit rates, traffic load, and many others. The benefits do not appear to significantly depend on the number of QAM modulators in the system. However, the number of QAM modulators that are needed to meet a blocking probability objective is highly dependent on the percentage of HD streams in the traffic mix. Future work includes the evaluation of the allocation algorithms in systems that have VoD, switched digital broadcast and high speed data (DOCSIS) [16] traffic using the same set of QAM resources. We also plan to develop models and tools that can be used for capacity planning in the next generation cable systems.

References

1. C. Aggarwal, J. Wolf, P. Yu, “On Optimal Batching Policies for Video-on-Demand Server”, ACM International Conference on Multimedia Systems, pp. 253-258, June 1996.
2. E. Coffman, M. Garey, D. Johnson, “Approximation Algorithms for Bin Packing: A Survey”, Approximation Algorithms for NP-hard Problems, pp 46-89, PWS Publishing Company, 1995.
3. T. Chiueh, C. Lu, “A Periodic Broadcasting Approach to Video-on-Demand Service”, Proc. SPIE, vol 2615, pp. 162-169, 1996.
4. A. Dan , D. Sitaram, P. Shahabuddin, “Scheduling Policies for an On-demand Video Server with Batching”, ACM International Conference on Multimedia, pp. 15-23 1994.
5. A. Dan, D. Sitaram, P. Shahabuddin, “Dynamic Batching Policies for an On-demand Video Server”, ACM Multimedia Systems, vol 4, pp. 112-121, 1996.
6. J. Flint, “ Marketers Should Learn to Stop Worrying and Love the PVR”, The Wall Street Journal, Oct 2005.
7. L. Gao, J. Kurose, D. Towsley, “Efficient Schemes for Broadcasting Popular Videos”, NOSSDAV 98, July 1998.
8. L. Golubchik, C. Lui, R. Muntz, “Adaptive Piggybacking: A Novel Technique for Data Sharing in Video-on-Demand Storage Servers”, ACM Multimedia Systems, vol. 4, no#0, pp 14-55, 1996.
9. J. Gong, Y. Syed, “Optimal QAM Assignment in the Presence of Mixed SD and HD Stream”, NCTA NationalShow 2005.
10. G. Hardin, “Session Resource Management: How to Slice the Pie Allocating Bandwidth for Standard and High-Def VOD”, Communications Technology Magazine, May 2005, available at : http://www.ct-magazine.com/archives/ct/0505/0505_sessionresource.htm

11. K. Hua, Y. Cai, S. Sheu, "Patching: A Multicast Technique for True Video-on-demand", IEEE Multimedia, vol. 4, pp. 51-62, 1997.
12. L. Juhn, L. Tseng, "Harmonic Broadcasting for Video-on-demand Service", IEEE Transactions on Broadcasting, vol 43 pp.268-271, Sept 1997.
13. W. Liao, V. Li, "The Split and Merge Protocol for Interactive Video-on-Demand", IEEE Multimedia, vol. 4, pp.51-62, 1997.
14. S. Lau, J. Lui, L. Golubchik "Merging Video Streams in a Multimedia Storage Server: Complexity and Heurisitcs", Multimedia Systems, vol. 6, no. 1, pp29-42, 1998.
15. S. Ross, "Introduction to Probability Models", Academic Press, 2003.
16. DOCSIS® Specifications, Cable Television Laboratories, Inc.(http:// www.cablemodem.com/ primer/)

Appendix A: Non-mixing Algorithm

In this appendix, we show the details of the non-mixing algorithm, taking a SD stream request as an example. The mathematical notations are defined in Section 4.

```
1. Identify a set of $I$ , s.t. $Q - q_i \geq r_S$ for $\forall i,\ i \in I$
        1.1 If $I$ is empty, reject the stream request;
2. Identify a subset of $J$ , $J \subseteq I$ , s.t. $S_j(q_j) = 3,\ j \in J$ ;
        2.1 If $J$ is empty, go to the next step;
        2.2 If $J$ has multiple elements, select $j^* = \arg \underset{j \in J}{Min}\ Q\text{-}q_j$ ;
        2.3 If there are multiple $j^*$ , select randomly among $j^*$ ;
3. Identify a subset of $J$ , $J \subseteq I$ , s.t. $S_j(q_j) = 1,\ j \in J$ ;
        3.1 If $J$ is empty, go to the next step;
        3.2 If $J$ has multiple elements, select $j^*$ randomly;
4. Identify a subset of $J$ , $J \subseteq I$ , s.t. $S_j(q_j) = 2,\ j \in J$ ;
        4.1 If $J$ is empty, go to the next step;
        4.2 If $J$ has multiple elements, select $j^* = \arg \underset{j \in J}{Min}\ Q\text{-}q_j$
        4.3 If there are multiple $j^*$ , select randomly among $j^*$ ;
5. Identify a subset of $J$ , $J \subseteq I$ , s.t. $S_j(q_j) = 4,\ j \in J$ ;
        5.1 If $J$ has multiple elements, select $j^* = \arg \underset{j \in J}{Max}\ Q\text{-}q_j$ ;
        5.2 If there are multiple $j^*$ , select randomly among $j^*$ ;
```

Performance of Experience-Based Admission Control in the Presence of Traffic Changes[⋆]

Jens Milbrandt, Michael Menth, and Jan Junker

Dept. of Distributed Systems, Inst. of Computer Science, University of Würzburg,
Am Hubland, 97072 Würzburg, Germany
Phone: +49 931 8886631; Fax: +49 931 8886632
{milbrandt, menth, junker}@informatik.uni-wuerzburg.de

Abstract. This paper investigates the transient behavior of *experience-based admission control* (EBAC) in case of traffic changes. EBAC is a robust and resource-efficient admission control (AC) mechanism used for reservation overbooking of link capacities in packet-based networks. Recent analyses gave a proof of concept for EBAC and showed its efficiency and robustness through steady state simulation on a single link carrying traffic with constant properties. The contribution of this paper is an examination of the memory from which EBAC gains its experience and which strongly influences the behavior of EBAC in both, stationary and non-stationary state. For the latter, we investigate the transient behavior of the EBAC mechanism through simulation of strong traffic changes which are characterized by either a sudden decrease or increase of the traffic intensity. Our results show that the transient behavior of EBAC partly depends on its tunable memory and that it copes well with even strongly changing traffic characteristics.

1 Introduction

Internet service providers operating next generation networks (NGNs) are supposed to offer quality of service (QoS) to their customers. As packet-based Internet protocol (IP) technology becomes more and more the basis of these networks, QoS in terms of limited packet loss, packet delay, and jitter is required to support real-time services. There are two fundamentally different methods to implement QoS: capacity overprovisioning (CO) and admission control (AC). With CO the network has so much capacity that congestion becomes very unlikely [1, 2], but this also implies that its utilization is very low even in the busy hour. Although CO is basically simple, it requires traffic forecasts and capacity provisioning must be done on a medium or long time scale.

In contrast, AC works on a smaller time scale. It grants access to flows with QoS requirements if the network load is sufficiently low and rejects excessive flow requests to shelter the network from overload before critical situations can

[⋆] Parts of this work were funded by the Bundesministerium für Bildung und Forschung of the Federal Republic of Germany (Förderkennzeichen 01AK045) and Siemens AG, Munich, Germany. The authors alone are responsible for the content of the paper.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 281–292, 2006.

occur. QoS is thus realized by flow blocking during overload situations. Compared to CO, AC requires less capacity and yields better resource utilization at the expense of more signaling, coordination and state management [3, 4, 5] especially in the context of a network-wide AC. In this work, we focus on link-based AC, i.e., on methods that protect a single link against overload. These methods are usually extended for application in entire networks. There are various approaches towards AC that can coarsely be classified into parameter-based AC (PBAC) [6, 7, 8], measurement-based AC (MBAC) [9, 10, 11, 12, 13, 14], and derivatives thereof.

PBAC, also known as (a priori) traffic-descriptor-based AC, is an approach appropriate for guaranteed network services [15], i.e., for traffic with imperative QoS requirements. It relies solely on traffic descriptors that are signaled by a source/application and describe the traffic characteristics of a flow such as the mean and the peak rate together with token bucket parameters. If an admission request succeeds, bandwidth is reserved and exclusively dedicated to the new flow. As a consequence, PBAC is often inefficient regarding its resource utilization since the traffic descriptors usually overestimate the actual rate to avoid traffic delay and loss due to spacing or policing. With PBAC, traffic is limited either by deterministic worst case considerations like network calculus [7, 8] or by stochastic approaches such as effective bandwidth [16]. In addition, PBAC calculations for heterogeneous traffic mixes can be very complex.

MBAC, in contrast, is an AC method adequate for controlled load network services [17], i.e., for traffic with less stringent QoS requirements. It measures the current link or network load in real-time and takes an estimate of the new flow to make the admission decision. The determination of traffic characteristics is thus shifted from a source/application to the network and the specified traffic descriptor, e.g. the peak rate, can be very simple. Most MBAC methods presented in the literature are link-oriented. They measure the current load on a link [11, 12, 13] while others perform measurements of individual flows [9]. Other MBAC approaches, also known as endpoint or probe-based AC [18, 19, 20], work on end-to-end measurements in terms of active or passive probing. All those MBAC methods take advantage of real-time measurements and admit traffic as long as enough capacity is available. The disadvantage of MBAC is its sensitivity to measurement accuracy and its susceptibility to traffic estimation errors which can occur, e.g., during QoS attacks, i.e. when admitted traffic flows are "silent" at the moment and congest the link/network later by simultaneously sending at high bitrate.

There are also hybrid AC approaches, e.g. [21], which combine techniques of PBAC and MBAC in a single AC framework to join the advantages of both AC schemes, i.e., they try to improve the network resource utilization while keeping the packet loss ratio below certain limits. However, these hybrid approaches still rely on real-time measurements or use rather static traffic estimators.

To the best of our knowledege, our method called *experience-based admission control* (EBAC) is the first hybrid AC approach that makes traffic measurements without real-time requirements and uses historical information about previously admitted traffic to make current admission decisions. The EBAC method was first introduced in [22]. With EBAC, a new flow is admitted to a link at time t if its peak rate together with the peak rates of already admitted flows does not exceed the link capacity multiplied by an overbooking factor $\varphi(t)$. The overbooking factor is calculated based on the reservation utilization of the admitted flows in the past. Hence, this method relies on experience. EBAC also requires traffic measurements to compute the reservation utilization. However, these measurements do not have real-time requirements and influence the admission decision only indirectly. A proof of concept for EBAC is given in [23] by simulations and corresponding waiting time analyses of the admitted traffic. In particular, EBAC has been investigated in steady state for traffic with rather static characteristics. MBAC methods are susceptible to traffic changes. Therefore, we investigate in this paper the behavior of EBAC in case of sudden traffic changes and its impact on the EBAC-controlled traffic.

The remainder of this paper is organized as follows. In Section 2, we briefly review the EBAC concept. Section 3 describes our simulation design and the applied traffic model and summarizes related results from previous work. In Section 4, we investigate the behavior of EBAC in case of sudden traffic changes. Section 5 summarizes this work and gives a conclusion.

2 Experience-Based Admission Control (EBAC)

In this section, we briefly review the EBAC concept with emphasis on the EBAC memory which holds the experience used to make AC decisions.

The idea of EBAC is briefly described as follows. An AC entity limits the access to a link l with capacity $c(l)$ and records all admitted flows $f \in \mathcal{F}(t)$ at time t together with their requested peak rates $\{r(f) : f \in \mathcal{F}(t)\}$. When a new flow f_{new} arrives, it requests a reservation for its peak rate $r(f_{new})$. If

$$r(f_{new}) + \sum_{f \in \mathcal{F}(t)} r(f) \leq c(l) \cdot \varphi(t) \cdot \rho_{max} \tag{1}$$

holds, admission is granted and f_{new} joins $\mathcal{F}(t)$. If flows terminate, they are removed from $\mathcal{F}(t)$. The experience-based overbooking factor $\varphi(t)$ is calculated by statistical analysis and indicates how much more bandwidth than $c(l)$ can be safely allocated for reservations. The maximum link utilization threshold ρ_{max} limits the traffic admission such that the expected packet delay W exceeds an upper delay threshold W_{max} only with probability p_W. Details regarding its computation are described in [23].

For the calculation of the overbooking factor $\varphi(t)$, we define the reserved bandwidth of all flows as $R(t) = \sum_{f \in \mathcal{F}(t)} r(f)$ while $C(t)$ denotes the unknown mean rate of the traffic aggregate $\mathcal{F}(t)$. EBAC makes traffic measurements $M(t)$ on the link and collects a time statistic for the reservation utilization $U(t) = M(t)/R(t)$. $U_p(t)$ is the p_u-percentile of the empirical distribution of U and the reciprocal of this percentile is the overbooking factor $\varphi(t) = 1/U_p(t)$. For the computation of the overbooking factor $\varphi(t)$ the following three functional components of the EBAC system are required:

1. *Measurement Process for* $M(t)$: To obtain $M(t)$, we use disjoint interval measurements such that for a time interval I_i with length Δ_i, the measured rate $M_i = \frac{\Gamma_i}{\Delta_i}$ is determined by metering the traffic volume Γ_i sent during I_i.

2. *Statistic Collection* $P(t, U)$: The reservation aggregate $R(t)$ is known from the AC process. The utilizations $U(t)$ are sampled in constant time intervals and are stored as hits in bins of a time-dependent histogram $P(t, U)$. The time-dependent utilization quantile $U_p(t)$ can be derived from $P(t, U)$ by
$$U_p(t) = \min_u \{u : P(t, U \leq u) \geq p_u\}. \tag{2}$$

3. *Statistic Aging Process*: If the traffic characteristics of the traffic aggregate $\mathcal{F}(t)$ change over time, the statistic collection $P(t, U)$ must forget obsolete data to increase the importance of the properties of the new traffic mix. Therefore, we record new samples by incrementing the respective bins by 1 and devaluate the contents of the histogram bins in regular devaluation intervals I_d by a constant devaluation factor f_d.

The histogram $P(t, U)$ implements the EBAC memory and the statistic aging process makes this memory forget about reservation utilizations in the past. The devaluation interval I_d and factor f_d yield typical half-life periods T_H after which collected values have lost half of their importance in the histogram. Therefore, we have $\frac{1}{2} = f_d^{T_H/I_d}$ and define the EBAC memory length based on the half-life period
$$T_H(I_d, f_d) = I_d \cdot \frac{-ln(2)}{ln(f_d)} \tag{3}$$
of the importance of reservation utilization values stored in the histogram. Various combinations of parameters (I_d, f_d) can lead to the same half-life period.

3 EBAC Performance Simulation

In this section, we first present the simulation design of EBAC on a single link and the traffic model we used on the flow and packet scale level. Afterwards, we summarize recent simulation results of EBAC in steady state.

3.1 Simulation Design

The design of our simulation is shown in Fig. 1. Different types of traffic *source generators* produce flow requests that are admitted or rejected by the *admission control* entity. To make an admission decision, this entity takes the overbooking factor $\varphi(t)$ into account. In turn, it provides information regarding the reservations $R(t)$ to the *EBAC system* and yields flow blocking probabilities $p_b(t)$. For each admitted source, a *traffic generator* is instantiated to produce a packet flow

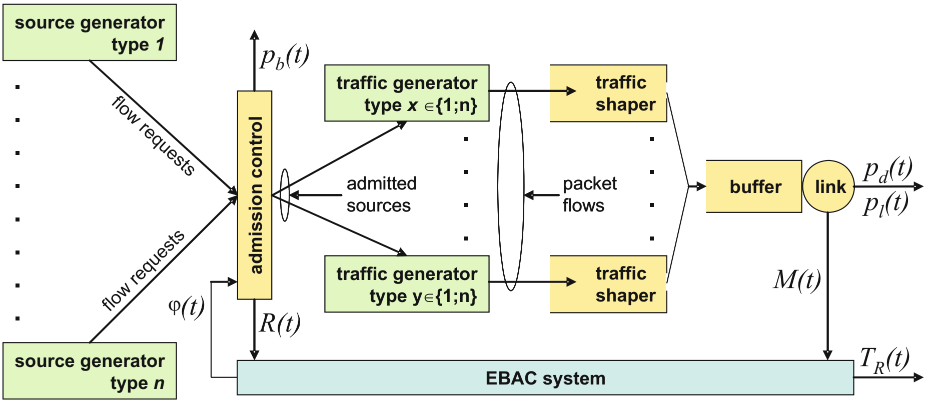


Fig. 1. Simulation design for EBAC in steady and transient state

that is shaped to its contractually defined peak rate. Traffic flows leaving the *traffic shapers* are then multiplexed on the buffered link with capacity $c(l)$. The link provides information regarding the measured traffic $M(t)$ to the EBAC system and yields packet delay probabilities $p_d(t)$ and packet loss probabilities $p_l(t)$. The primary performance measure of our non-stationary EBAC simulations is the overall response time T_R, i.e., the time span required by the EBAC system to fully adapt the overbooking factor to a new traffic situation.

3.2 Traffic Model

In our simulations, the admission-controlled traffic is modelled on two levels, namely the flow scale level and the packet scale level. While the flow level controls the inter-arrival times of flow requests and the holding times of admitted flows, the packet level defines the inter-arrival times and the sizes of packets within a single flow.

Flow Level Model. On the flow level, we distinguish different traffic source types, each associated with a characteristic peak-to-mean rate ratio (PMRR, defined below) and corresponding to a source generator type in Fig. 1. The inter-arrival time of flow requests and the holding time of admitted flows both follow a Poisson model [24], i.e., new flows arrive with rate λ_f and the duration of a flow is controlled by rate μ_f, where $1/\mu_f$ denotes the mean flow holding time. For the non-stationary EBAC simulations, the source-type specific parameters λ_f vary over time which directly impacts the load and the composition of the traffic on the link. Provided that no blocking occurs, the overall offered load $a_f = \lambda_f/\mu_f$ is the average number of simultaneous flows measured in Erlang. If not mentioned differently, the holding time of a flow is exponentially distributed with a mean of $1/\mu_f = 90$ s and the traffic load is set to $a_f \geq 1.0$ such that the EBAC-controlled link is saturated with flow requests. The latter assumption allows for an investigation of the EBAC performance under heavy traffic load where some flow requests are rejected.

Packet Level Model. On the packet level model, we abstract from the wide diversity of packet characteristics induced by applications and different transmission protocols. Since we are interested in the basic understanding of the transient behavior of EBAC, we abstract from real traffic patterns and define a flow of consecutive data packets simply by a packet size and a packet inter-arrival time distribution. Both contribute to the rate variability within a flow that is produced by a traffic generator in Fig. 1. To keep things simple, we assume a fixed packet size and use a Poisson arrival process to model a packet inter-arrival time ditribution with rate λ_p. We are aware of the fact that Poisson is not a suitable model to simulate Internet traffic on the packet level [25]. We therefore don not take it unconditioned, but use it for the generation of packet streams that are subsequentially policed by peak-rate traffic shapers (cf. Fig. 1). The properties of the flows are primarily determined by the configuration of these shapers. In practice, the peak rate $r(f)$ of a flow f is limited by an application or a network element and the mean rate $c(f)$ is often unknown. In our simulations, however, the mean rate is known a priori and we control the rate of flow f by its peak-to-mean rate ratio $k(f) = r(f)/c(f)$. Analogously, $K(t) = R(t)/C(t)$ is the peak-to-mean rate ratio of the entire traffic aggregate $\mathcal{F}(t)$ at time t. It is a natural upper limit for the achievable overbooking factor $\varphi(t)$.

3.3 EBAC in Steady State

The intrinsic idea of EBAC is the exploitation of the peak-to-mean rate ratio $K(t)$ of the traffic aggregate admitted to the link. In [23], we simulated EBAC on a single link with regard to its behavior in steady state, i.e., when the properties of the traffic aggregate were rather static. These simulations provided a first proof of concept for EBAC. We showed for different peak-to-mean rate ratios that EBAC achieves a high degree of resource utilization through overbooking while packet loss and packet delay are well limited. Further simulation results allowed us to give recommendations for the EBAC parameters such as measurement interval length and reservation utilization percentile to obtain appropriate overbooking factors $\varphi(t)$. They furthermore showed that the EBAC mechanism is robust against traffic variability in terms of packet size and inter-arrival time distribution as well as to correlations thereof.

4 EBAC Performance Under Traffic Changes

This section discusses the transient behavior of EBAC when the characteristics of the EBAC-controlled traffic change significantly. We investigate the response time T_R of EBAC to provide a new appropriate overbooking factor $\varphi(t)$ after a decrease or increase of the traffic intensity. We consider sudden changes of the traffic characteristics to have worst case scenarios and to obtain upper bounds on the response time. We simulate them with only a single type of traffic flows since only the properties of the entire admitted traffic aggegate are of interest for the calculation of the overbooking factor.

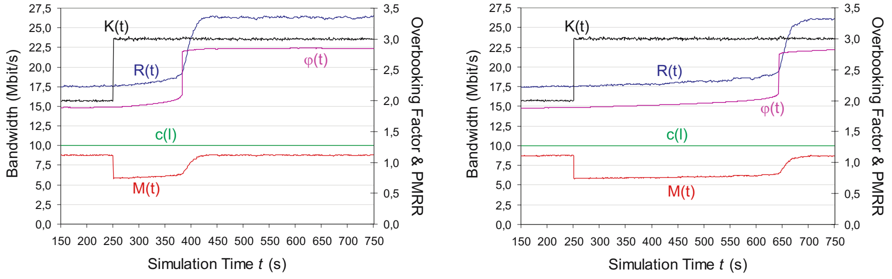
4.1 Decrease of the Traffic Intensity

We first investigate the change of the traffic intensity from a high to a low value which corresponds to an increase of the peak-to-mean rate ratio $K(t)$ of the entire traffic aggregate, i.e., all currently and future admitted traffic sources simultaneously reduce their sending rate. For all simulation experiments, we use a link capacity of $c(l) = 10$ Mbit/s and a reservation utilization quantile $p_u = 99\%$. On the flow level, we set the traffic characteristics $\lambda_f = \frac{1}{750 \text{ ms}}$ and $\mu_f = \frac{1}{90 \text{ s}}$ to offer enough traffic to the link, i.e., the link is saturated with traffic. At simulation time $t_0 = 250$ s, the peak-to-mean rate ratio suddenly increases from $K(t) = 2$ to $K(t) = 3$, i.e., all traffic sources slow down and use a lower packet arrival rate λ_p. Figures 2(a) and 2(b) illustrate simulations averaged over 50 runs for a short and a long EBAC memory. The sudden increase of the peak-to-mean rate ratio results in an immediate decrease of the consumed link bandwidth $M(t)$. As a consequence, the reservation utilization $U(t) = M(t)/R(t)$ also decreases. After a while, the histogram has collected enough low utilization values so that its 99%-percentile $U_p(t)$ decreases which, in turn, leads to a higher overbooking factor $\varphi(t)= 1/U_p(t)$. Hence, more traffic sources are admitted to the link and the reserved rate $R(t)$ rises. Finally, the EBAC system stabilizes again with an expected overbooking factor $\varphi(t) \approx 3$. The speed of the adaptation process is obviously influenced by the EBAC memory.

To measure the duration of the transient phase, i.e., the time until the overbooking factor reaches a new stable value, we calculate the difference between the peak-to-mean rate ratio $K(t)$ and the overbooking value $\varphi(t)$. If $K(t) - \varphi(t) < \varepsilon$, the transition between the two traffic scenarios is considered completed and the EBAC system is in steady state again. We therefore define the EBAC response time

$$T_R = \min\left\{t_i - t_0 : K(t_i) - \varphi(t_i) < \varepsilon \wedge t_i > t_0\right\} \tag{4}$$

and set the threshold $\varepsilon = 0.2$ in our simulations. This value is specific to our experiments but seems to be appropriate with regard to the asymptotic convergence of $\varphi(t)$ to $K(t)$. Due to the design of the EBAC mechanism, $\varphi(t) \leq K(t)$



(a) Short memory with half-life period $T_H = 20$ s (b) Long memory with half-life period $T_H = 60$ s

Fig. 2. Impact of the EBAC memory on the time-dependent overbooking performance

always holds. The statistical significance of our results is assured by calculating the 95%-confidence intervals of the overbooking factor $\varphi(t)$ within 50 iterations of the simulation. As a result, the confidence intervals turn out to be so narrow that we omit them in Fig. 2 for the sake of clarity.

The different progressions of the overbooking factor $\varphi(t)$ in Fig. 2(a) and 2(b) show that in this experiment, the EBAC response time T_R strongly depends on the EBAC memory length represented by the half-life period T_H. To investigate the correlation between T_R and T_H, we perform a series of experiments with varying half-life periods and measure the EBAC response times. Figure 3 shows that there is an almost linear dependency between the EBAC response time T_R and the half-life period T_H of the EBAC memory.

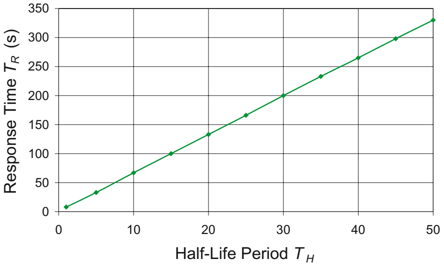


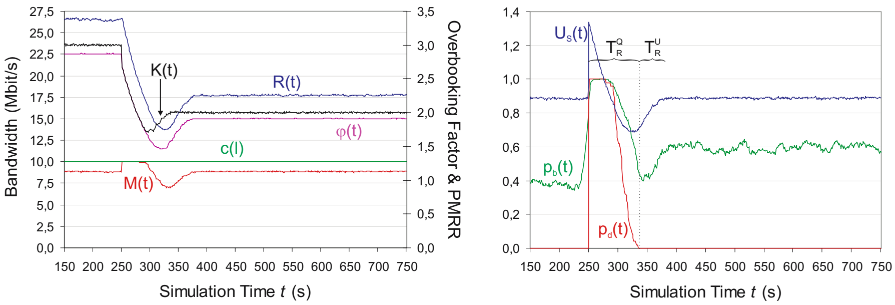
Fig. 3. Correlation of EBAC memory with half-life period T_H and EBAC response time T_R for decreasing traffic intensity

4.2 Increase of the Traffic Intensity

We now change the traffic intensity from a low to a high value which corresponds to a decrease of the peak-to-mean rate ratio $K(t)$ of the entire traffic, i.e., all admitted and future traffic sources raise their sending rate simultaneously which corresponds to a collaborative QoS attack. In contrast to the previous experiment, the QoS is at risk since the link suddenly gets overloaded and the packet delay and flow blocking probabilities increase as could be expected for a QoS attack. To blind out the impact of the link buffer on the EBAC response time, we set its value to infinity.

The QoS attack experiment is designed analogously to the decrease of the traffic intensity in Sec. 4.1. The only difference is that the packet arrival rates λ_p of all flows are increased such that the peak-to-mean rate ratio decreases from $K(t) = 3$ to $K(t) = 2$. Figures 4 and 5 show the overbooking and QoS performance of the EBAC system for short and long EBAC memory, respectively.

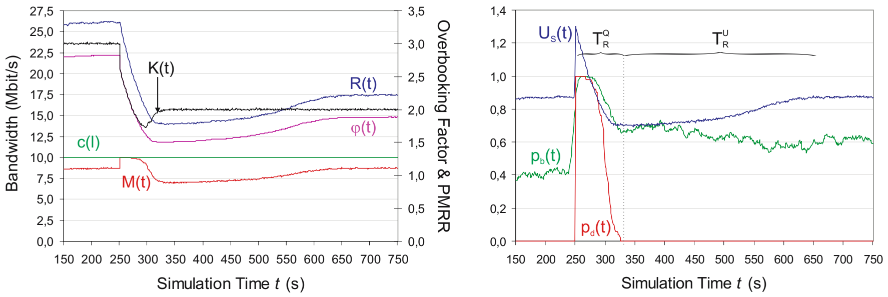
At time $t_0 = 250$ ms the QoS attack starts. As the link becomes overutilized, the fill level of the link buffer increases and the probability for excessive packet delay $p_d(t) = P(\text{packet delay} > 50\text{ms})$ and the flow blocking probability $p_b(t)$ raise to 100% (cf. Fig. 4(b) and 5(b)). As another consequence, the overbooking factor $\varphi(t)$ decreases due to a rising reservation utilization quantile $U_p(t)$ and all newly arriving flows are blocked by EBAC. As time goes by, some admitted



(a) Overbooking performance

(b) QoS performance

Fig. 4. Time-dependant EBAC performance during a QoS attack for a short EBAC memory with half-life period $T_H = 5.76$ s



(a) Overbooking performance

(b) QoS performance

Fig. 5. Time-dependant EBAC performance during a QoS attack for a long EBAC memory with half-life period $T_H = 65.79$ s

flows expire and their reserved bandwidth is released. However, the overbooking factor $\varphi(t)$ is further decreased as long as the packet delay and the link load are high. Hence, the overbooking factor decreases below its target value of $\varphi(t) \approx\ 2$ (cf. Fig. 4(a) and 5(a)). When enough flows have expired, the link buffer empties and the QoS is restored as a result of the decreased overbooking factor. Figures 4(b) and 5(b) show that the QoS recovery duration T_R^Q is almost the same for short and long EBAC memory, respectively. After a certain time span T_R^U, the overestimated reservation utilizations in the histogram are faded out by statistic aging. Simultaneously, the overbooking factor $\varphi(t)$ and the link utilization $U_S(t)$ converge to stable values when the EBAC system reaches its steady state again. In contrast to Equ. 4, we now define the EBAC response time as

$$T_R = T_R^Q + T_R^U \tag{5}$$

where interval $T_R^Q = \min\{t_i - t_0 : p_d(t_i) = 0 \wedge t_i > t_0\}$ and time span $T_R^U = \min\{t_j - (t_0 + T_R^Q) : K(t_i) - \varphi(t_i) < \varepsilon \wedge t_j > t_0 + T_R^Q\}$. Our simulation results

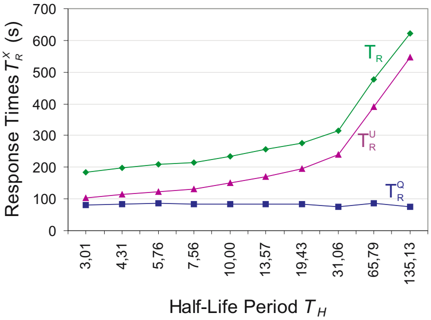


Fig. 6. Correlation of EBAC memory with half-life period T_H and EBAC response times T_R^X for increasing traffic intensity

compiled in Fig. 6 show that the EBAC memory length represented by the half-life period T_H influences its overall response time T_R after a QoS attack. However, it does not influence the time T_R^Q that is required to recover the QoS. We conclude that the EBAC memory length influences the adaptation speed of the overbooking factor only for a decrease of the traffic intensity. Hence, the parameter T_H provides a means for configuring the conservativeness of the EBAC system.

For the sake of completeness, we performed further QoS attack experiments to investigate the impact of the link buffer size, the mean flow holding time, and the link capacity on the transient behavior of EBAC. Details on these experiments are omitted due to the lack of space. We just summarize their results as follows: T_R^Q and T_R^U both depend on the buffer size B and the mean flow holding time $1/\mu_f$. Hence, larger buffers and longer flow holding times extend T_R. In contrast, the link capacity $c(l)$ has no effect on the overall EBAC response time T_R. The above statements hold for arbitrary settings of T_H.

5 Conclusion

In this paper, we investigated the transient behavior of experience-based admission control (EBAC) on a single link.

EBAC is a new link admission control paradigm [22] and represents a hybrid solution between parameter-based and measurement-based admission control. We briefly reviewed the EBAC system whose proof of concept was already given in [23] for traffic with stationary characteristics. We explained the simulation design and the traffic model used for the analyses of the transient behavior of EBAC. We finally simulated EBAC under extremely changing traffic conditions and showed the results which build the main contribution for this paper.

As EBAC partly relies on traffic measurements, it is susceptible to changes of the traffic characteristics. There are certain influencing parameters coupled with this problem. One of them is the length of the EBAC memory which has been defined by its half-life period T_H. We tested the impact of the EBAC memory on a sudden decrease and increase of the traffic intensity which has been expressed by the change of the peak-to-mean rate ratio of the simulated traffic flows. We showed that, for a changing traffic intensity, the response time T_R required to adapt the overbooking factor to the new traffic situation depends linearly on the half-life period T_H. For decreasing traffic intensity, the QoS of the traffic was not at risk. For a suddenly increasing traffic intensity, however, it was compromised for a certain time span T_R^Q which was less than the average flow holding time. Note that the respective experiment used an unlimited link buffer and investigated the performance of EBAC under very extreme traffic conditions that correspond to a collaborative and simultaneous QoS attack by all traffic sources.

Currently, we are working on an EBAC extension, called type-specific overbooking, that provides different overbooking factors for traffic from different applications. In future investigations, the performance of EBAC may be studied with real traffic traces and the concept may be extended to a network-wide scope.

References

1. Martin, R., Menth, M., Charzinski, J.: Comparison of Link-by-Link Admission Control and Capacity Overprovisioning. In: Proc. of 19th International Teletraffic Congress (ITC19), Beijing, China (2005)
2. Martin, R., Menth, M., Charzinski, J.: Comparison of Border-to-Border Budget Based Network Admission Control and Capacity Overprovisioning. In: Proc. of 4th International IFIP Networking Conference, Waterloo, Canada (2005)
3. Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: RFC 2205: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification (1997)
4. Yavatkar, R., Pendarakis, D., Guerin, R.: RFC 2753: A Framework for Policy-Based Admission Control (2000)
5. Yavatkar, R., Hoffman, D., Bernet, Y., Baker, F., Speer, M.: RFC 2814: SBM (Subnet Bandwidth Manager): A Protocol for RSVP-Based Admission Control over IEEE 802-Style Networks (2000)
6. Wroclawski, J.: RFC 2210: The Use of RSVP with IETF Integrated Services (1997)
7. Boudec, J.L.: Application of Network Calculus to Guaranteed Service Networks. IEEE Transactions on Information Theory **44**(3) (1998)
8. Fidler, M., Sander, V.: A Parameter Based Admission Control for Differentiated Services Networks. Computer Networks **44**(4) (2004) 463–479
9. Gibbens, R., Kelly, F.: Measurement-Based Connection Admission Control. In: Proc. of 15th International Teletraffic Congress, Washington D. C., USA (1997)
10. Jamin, S., Shenker, S., Danzig, P.: Comparison of Measurement-Based Call Admission Control Algorithms for Controlled-Load Service. In: Proc. of IEEE Infocom 2000. (1997) 973–980
11. Grossglauser, M., Tse, D.: A Framework for Robust Measurement-Based Admission Control. IEEE Transactions on Networking **7**(3) (1999) 293–309
12. Breslau, L., Jamin, S., Shenker, S.: Comments on the Performance of Measurement-Based Admission Control Algorithms. In: Proc. of IEEE Infocom. (2000)
13. Mandjes, M., van Uitert, M.: Transient Analysis of Traffic Generated by Bursty Sources, and its Application to Measurement-Based Admission Control. Telecommunication Systems **15**(3-4) (2000) 295–321

14. Qiu, J., Knightly, E.: Measurement-Based Admission Control with Aggregate Traffic Envelopes. IEEE Transactions on Networking **9**(2) (2001) 199–210
15. Shenker, S., Partridge, C., Guerin, R.: RFC 2212: Specification of Guaranteed Quality of Service (1997)
16. Roberts, J., Mocci, U., Virtamo, J.: Broadband Network Teletraffic - Final Report of Action COST 242. Springer, Berlin, Heidelberg (1996)
17. Wroclawski, J.: RFC 2211: Specification of the Controlled-Load Network Element Service (1997)
18. Cetinkaya, C., Knightly, E.: Egress Admission Control. In: Proc. of IEEE Infocom 2000. (2000) 1471–1480
19. Elek, V., Karlsson, G., Rönngren, R.: Admission Control Based on End-to-End Measurements. In: Proc. of IEEE Infocom 2000. (2000) 1233–1242
20. Más, I., Karlsson, G.: PBAC: Probe-Based Admission Control. In: 2^{nd} International Workshop on Quality of future Internet Services (QofIS 2001). (2001)
21. Georgoulas, S., Trimintzios, P., Pavlou, G.: Joint Measurement- and Traffic Descriptor-Based Admission Control at Real-Time Traffic Aggregation Points. In: Proc. of IEEE Int. Conference on Communications (ICC 2004), QoS and Performance Symposium, Paris, France (2004)
22. Milbrandt, J., Menth, M., Oechsner, S.: EBAC - A Simple Admission Control Mechanism. In: Proc. of 12^{th} IEEE International Conference on Network Protocols (ICNP 2004), Berlin, Germany (2004)
23. Menth, M., Milbrandt, J., Oechsner, S.: Experience Based Admission Control (EBAC). In: Proc. of 9^{th} IEEE Symposium on Computers and Communications (ISCC 2004), Alexandria, Egypt (2004)
24. Law, A.M., Kelton, W.D.: Simulation Modeling and Analysis. McGraw-Hill, Boston, USA (2000)
25. Paxson, V., Floyd, S.: Wide-Area Traffic: The Failure of Poisson Modeling. IEEE/ACM Transactions on Networking **3**(3) (1995) 226–244

Topologically-Aware AAA Overlay Network in Mobile IPv6 Environment*

Jun Li[1,2,3], Xin-ming Ye[2], and Ye Tian[1,3]

[1] Institute of Computing Technology, Chinese Academy of Sciences,
No. 6 Kexueyuan South Avenue, Beijing, 100080, China
[2] Department of Computer Science, Inner Mongolia University,
No. 235 Daxue Avenue, Hohhot, 010021, China
[3] Graduate University of Chinese Academy of Sciences,
No. 19 Yuquan Avenue, Beijing, 010021, China
{lijun, jack_ty}@ict.ac.cn, xmy@imu.edu.cn

Abstract. In mobile IPv6 network, AAA mechanism is necessary for administration and security because roaming nodes are permitted and become majority. However, disharmonies are exposed when MIPv6 meets AAA. On one hand, AAA procedures increase the latency of MIPv6 handover by inserting several message round trips before mobile registration. Thus the handover performance is reduced. On the other hand, AAA does nothing to help MIPv6 with its security problem. The fact is that MIPv6 has to struggle with those problems by itself. The result is that MIPv6 become complicated and inefficient. The crux of the matter is that AAA and MIPv6 are separately designed from their own viewpoints without mutual reinforcement. In this study, a Topologically-Aware AAA Overlay Network (TA^{4}ON) is used for compatibly combining together resources and capacities from both sides. All connected AAA participators construct an overlay network which is naturally topology-aware. MIPv6 security issues, for example key generating and peer identifying, are handled by AAA. Secret materials and even MIPv6 signals can be delivered through TA^{4}ON. As shown in this paper, at little additional cost all things serve their proper purposes and finally performance and security of MIPv6 are improved.

Keywords: Mobile IPv6, AAA, Overlay Network, Performance, Security.

1 Introduction

At the beginning of defining Mobility support in IPv6 (MIPv6) [1] , researchers came to an agreement that performance and security are very important for it. Almost at the same time when MIPv6 specification was a draft, several enhancement solutions were discussed constructively. Among them, some enhancements were to improve the handover performance. The most famous of them are FMIPv6 [2] and HMIPv6 [3], they became experimental standards recently.

Enhancements for security were proposed, too. IPSec [4][5] was used to protect the communication between Mobile Node (MN) and Home Agent (HA) [6]. Also, that

* This work was supported by NSFC grant No. 60263002.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 293–306, 2006.

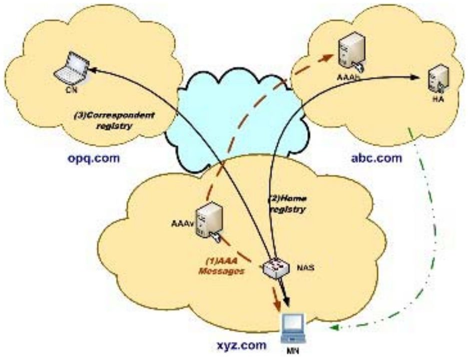


Fig. 1. AAA mechanism and MIPv6 handover. *Abc.com is the home domain of MN and AAAh is its registry AAA server. Xyz.com is the current domain where MN is located. In domain xyz.com, NAS is a Network Access Server and AAAv is the designated AAA server for that NAS.*

proposal became a standard and was released with MIPv6. Several other proposals are under discussed, for an instant, [7] is for authenticating MN while it is away from home. However, In MIPv6, only internal security problems are cared. Some problems are resolved locally without an overall consideration.

From the viewpoint of administration and security, Authentication, Authorization and Accounting (AAA) mechanism is foundational infrastructure for the entire network. Almost all the operators have deployed AAA in their networks, and it will be the same situation in the future mobile Internet. AAA infrastructures are connected with each other. Communications between them are protected by pre-shared Security Association (SA). AAA infrastructures that belong to different administrative domains or different operators can be connected to provide service to subscribers of other domains [8]. Of course, in this case administrators involved must have reached an agreement in advance.

Here comes the problem. On one hand, AAA delays MIPv6 handover by inserting the access control before mobile registration procedure.

See figure 1. MN has to perform home registration (step 2 in figure 1) and correspondent registration (step 3 in figure 1) only after it has been authenticated and authorized to access network by back end AAA server (step 1 in figure 1). Authentication procedure may be very time-consuming according to network topology. Particularly, when MN is in a foreign administrative domain, authentication messages may have to pass through several administrative networks and AAA servers to get to its destination.

Because the total handover latency is nearly doubled comparing to the situation without access control, time sensitive applications will not work well. This is a real challenge for MIPv6. HMIPv6 can't keep away from this difficulty either. FMIPv6 gets even worse in that it can't be "fast" and just works as standard MIPv6. To address such issue, a good idea is to combine two procedures together, ie. integrats home registration messages into AAA messages and let home AAA server communicate with HA [9][10]. However, it is not enough only to care the home registration procedure. Correspondent registration should be taken into account, too. Some other solutions [11][12] were proposed just to give architecture for MIPv6 and AAA to work together but handover performance was not considered.

On the other hand, security is another focal point of MIPv6 handover. However, AAA does nothing for it though AAA is good at this.

MIPv6 is trying hard to resolve all problems by itself, including security issues. For example, [7] is trying to authenticating MN by carrying authentication data in packet header. At the same time, instead of resolving all the security issues listed in [1] and [13], this will make MIPv6 so complicated, unoperationable and inefficient. Return Routability (RR) procedure is used in [1] to protect correspondent registration, however, there are still many flaws in it as listed in [1]. [14] is trying to secure the route optimization between MN and CN by static key, however there are still some limits to use its mechanism, such as MN and CN must be in the same domain and so on. This is not flexible enough for future global mobility.

[15] gave a solution to protect all traffic for MIPv6, but HA is its bottleneck. It's already a heavy burden for HA to record the positions and forward messages for all roaming MNs. That solution can make HA overworked. Under this situation, HA becomes a vulnerable link of the entire chain.

On the contrary, we know that AAA infrastructures have powerful computing capacity and are natively good at security concerns. In addition, AAA servers can be connected to construct an overlay network, so they know easily the topology of network in a large scale. They should do more good to MIPv6 rather than just carry home registration messages. So far, we see few studies focusing on this point.

AAA and MIPv6 have been developed separately. Both of them have their own purposes, protocols, mechanisms, application field and so on. It's not easy to combine them together and work harmoniously. We are trying to resolve this issue in a novel way. From the point of view of our study, AAA infrastructures are the "virtual backbone" of inter-networks at application layer, and they are core layer for administration and security.

In light of those situations mentioned above, this research is carried out to achieve three goals below,

- *Compatibly merge AAA mechanism and Mobile IPv6;*
- *Speed up signal delivery for Mobile IPv6;*
- *Provide security service for Mobile IPv6.*

The main contribution of this study is (1) TA^{4}ON is used to enhance MIPv6. Performance and security level are both improved; (2) Only little additional costs are added to AAA infrastructures and MIPv6 remains specialized and efficient.

The rest of this paper is organized as follows. Section 2 introduces the background of this study. Section 3 describes the framework of TA^{4}ON. Section 4 shows in detail how TA^{4}ON enhances MIPv6. Section 5 gives a analysis and evaluation. Section 6 ends this paper with conclusion and future work.

2 Preliminary

In this section, Mobile IPv6 and Diameter base protocol (the newest and typical version of AAA protocol) are described as background of this research respectively. The significance of performance and security for MIPv6 is highlighted. The topologically-aware inter-connection on application layer is described as the feature of Diameter.

2.1 Mobility Support in IPv6 (MIPv6)

IPv6[16], above all, has a huge address space which is believed never to be exhausted. MIPv6 is fresh blood to IPv6 family. It adds feature of host mobility to IPv6.

In figure 2, a MIPv6-enabled Node (MN) has a permanent address at its registry network. That address is called Home Address (HoA) and the registry network is *home network*. Networks except home network are all called *foreign networks*. Each time MN handovers to or boots at a foreign network, it will get a Core-of Address (CoA) by stateful or stateless address auto-configuration① [17]. The subnet prefix of CoA is same as that foreign network's.

There is at least one important fixed node, called Home Agent (HA), at home network. Every time MN gets a new CoA (the new one is called Current CoA, CCoA. The old one is called Previous CoA, PCoA.), it must register that CCoA with its HoA to HA by exchanging binding messages②. That procedure is called Home Registration (HR). HA's functionality is to impersonate MN by proxy neighbor discovery when MN is away from its home network. It receives packets destined MN's HoA③, then forward them to the CCoA of MN via a pre-established tunnel④ and reversely. All communications between HA and MN are protected by IPSec with a pre-shared Security Association (SA) [6]. At this point, MN can be reached again after changing its point of attachment.

MN may optionally send binding update message⑦ to the source of those packets, called Correspondent Node (CN), to eliminate triangle routes, MN-HA-CN. That procedure is called Correspondent Registration (CR). Those binding messages are protected by a leading Return Routability (RR) procedure ⑤⑥ (dashed line in figure 2). Then CN and MN can send message directly to each other⑧ without detour via HA. Usually this is called route optimization.

Procedure HR, RR and CR are performed one after the other. Messages exchanged during HR, RR and CR are called *handover signal messages*. HR messages are prerequisite for handover and it must be the first step among them. CR is carried out only for the purpose of route optimizing after HR is completed successfully. RR is the leading security procedure for CR.

So as far as performance is concerned, HR is the first step to recover the reachability. However, HR is not enough in most cases because reachability is not really recovered

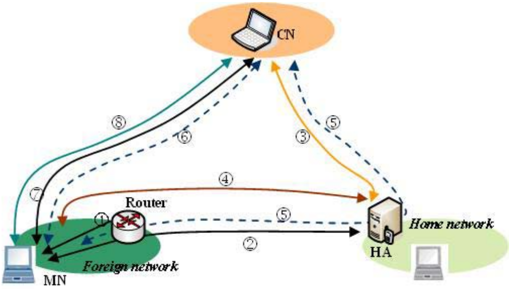


Fig. 2. Communications while MN moves to a foreign link

immediately after HR. Usually CN insists to send packets to MN's PCoA before CR is finished. If CR is late or somehow aborted, then CN has to sent packets to MN's home network. At this time, reachability is really restored. Even so, too many packets have been lost and HA may become the bottle neck of route. Again if somehow HR is late, then the situation becomes worse because even HA does not know at all where MN is. Granting that everything goes as our wishes, "optimized route" may be the worse one because it is not always the truth in a Internet route triangle that the sum cost of two lines is greater than the cost of the third one [18]. Usually applications require low latency time and small rate of packet lose on handover, so handover performance is a challenge for MIPv6.

Security is another coin side for MIPv6. To protect communication between MN and HA, including HR messages, IPSec with pre-shared SA is used[6]. But pre-shared SA is not flexible enough for MIPv6, for example, to support Dynamic Home Agent Address Discovery (DHAAD) and Mobile Prefix Discovery (MPD). DHAAD and MPD in MIPv6 may leak information about network topology and make MN to be tricked into believing false information about prefixes. Although CR is protected by a leading RR procedure, threads still exist because RR is not flawless. Security problems of MIPv6 are listed in [1]. Nevertheless, no applicable solution is given.

In some place, security procedures conflict with handover performance. For example, RR procedure defers CR procedure at least 1.5 round trips. If IKE[19] is used for generating dynamic SA, then SA has to be regenerated on each time handover, which must detain HR for a few round trips. MIPv6 is facing a dilemma of emphasizing performance or security.

2.2 Diameter Based AAA Mechanism

MIPv6-capable mobile nodes can roam among networks that belong to their home Service Provider as well as others. This is a result of the service level agreements that exist between operators. One of the key AAA protocols that allow this kind of roam mechanism is Diameter [8]. [9] is an Internet Draft specifies a new application to Diameter that supports Mobile IPv6.

According to Diameter base protocol, Diameter connections are established between each two adjacent AAA servers as well as between AAA servers and their own subordinate NASs as soon as they begin to serve. Communications are protected by pre-shared SA. Connections will be kept until service stops. *So we can believe that connections are always secure and stable.* The methods for connection setup and communication protection are beyond the scope of this paper.

In figure 3, there are three administrative domains which may belong to different operators. Among those operators roaming agreements exist. In each domain there may be several networks or sub-networks, and at least one AAA server. AAA1, AAA2 and AAA3 are Diameter AAA servers located at different domains and serve their own subscribers respectively. For example, AAA1 is the registry server of host1 at its home domain, "xyz.com". And AAA2 is the registry server of host2 at its home domain, "opq.com". Network Access Server (NAS) is the command executor for the designated AAA server. It is the nearest AAA infrastructure to hosts which need to access network.

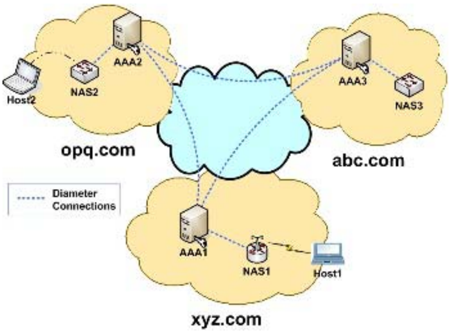


Fig. 3. Diameter based AAA mechanism

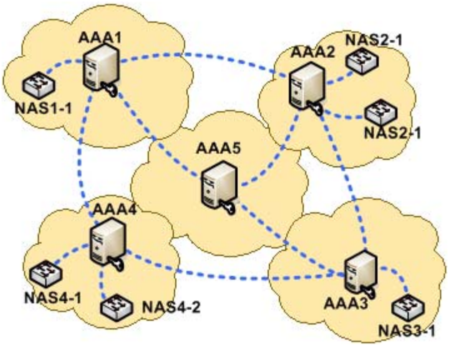


Fig. 4. Diameter based connections in a large scale

Next we will describe a typical occasion of AAA message flow. If, sometime NAS1 has to authenticate a host registered at a foreign domain, say abc.com, a request message originated from that host is forwarded to a designated AAA server, AAA1. And then some messages are exchanged between AAA1 and AAA3. If necessary, messages may pass an inter-mediate domain, and then an AAA server in that domain is responsible for forwarding the messages forth and back. When a result message eventually arrives at NAS1, it will know how to do with that host.

Figure 4 shows a bigger inter-network. There are five connected Diameter AAA servers and their own five subordinate administrative domains. There are immediate connections among some of them. AAA4 and AAA5 can be bridges for AAA1 and AAA3, AAA1 and AAA3 can be bridges for AAA2 and AAA4. However, those candidates may not become real bridges. Whether a AAA server is a bridge for others is decides by administrators of involved domains. They must take necessity, cost, security and performance into consideration. A real AAA bridge server must be the result of agreeing on those factors.

3 Topologically-Aware AAA Overlay Network

In our opinion, AAA infrastructures connected together via Diameter protocol are switch nodes of an overlay network. Let's take a new look at figure 3. If only AAA messages are considered, then an overlay network can be observed. AAA servers are switch nodes and AAA connections are links between two adjacent switch nodes. If a domain is big enough then AAA servers in it can be connected hierarchically to obtain high efficiency of management.

Furthermore, this AAA overlay network is born topologically-aware. Usually AAA servers and NASs are deployed along the topology of networks. Administration domain is composed of several physical networks (or subnetworks), each of which there is at least one NAS working for a designated AAA server which is in charge of businesses of this domain. Diameter connections between different domain are setup between two AAA servers located in top level of each domain respectively.

If we think about this overlay network in a much larger scale, then it becomes "virtual backbone" for the whole inter-network. Top level AAA servers are core switch nodes and NASs become the edge switch nodes of this overlay network (See figure 4). Again if we make AAA infrastructures to shoulder much more responsibility, then they will be the "core layer" of administration for the whole inter-network. This is exactly what we are doing and the object is called Topologically-Aware AAA Overlay Network (TA^{4}ON).

3.1 Basic Assumptions

Before going on the discussion, we have following basic assumptions. Some of them were mentioned above, but we reorganize them below for clearly understanding.

1) The scope of operation may cover several administrative domains which may belong to different operators. There are agreements among them to grant AAA servers to be connected together using AAA protocol like Diameter.
2) There is at least one AAA server in an administrative domain. AAA servers and Network Access Servers in a same domain are connected to be a hierarchical structure.
3) At least one AAA server (usually the one at top level) in each domain is the Gateway Server (GS). GSs are interconnected for exchanging and forwarding AAA messages according to agreements between operators.
4) The communication between each two connected AAA servers and between AAA servers and NASs is protected in some way. It is believed secure to communicate through those connection. However, how to do so is out of the scope of this paper.
5) Administration factors described in subsection 2.2 are ignored. A Diameter AAA server is a bridge server for others only if necessary and possible.

3.2 Framework

To achieve the objectives listed in section 1, we want the AAA overlay network to delivery as many as MIPv6 signals. And it must get much more information about the inter-network, such as topology and status.

Figure 5 is a diagram for our framework of TA^{4}ON. To simplify our discussion, there is only one AAA server in each of those three administrative domains. So they are all GSs and are interconnected to be an AAA overlay network. Not only MN but also HA and CN are connected to this overlay network. They all trust the nearest AAA infrastructures (AAA server and NAS) in their registry domain, then indirectly the AAA infrastructures in other domains. Each node has a unique identity for living in this overlay network. Usually Network Access Identity (NAI) [20] or Mobile Node Identity (MNI) [21] is used. NAI looks like someone@somedomain.com, where somedomain.com is the name of registry domain.

MN, CN and HA need to establish trust relationships with AAA infrastructures in their registry domain. They are also required to understand new type of AAA messages and the MIPv6 signals inside.

Note that only AAA messages run on the AAA overlay network, including original and new created types. Other messages, such as MIPv6 regular signals and normal data stream, are transported as usual.

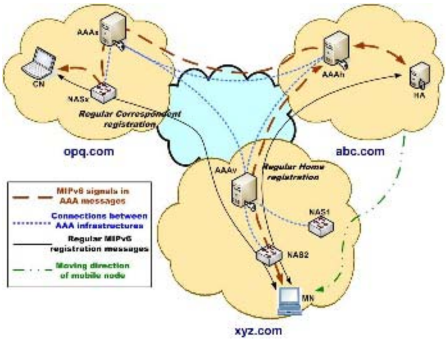

AAAx
CN
NASx
Regular Correspondent registration
AAAh
HA
opq.com
abc.com
AAAv
Regular Home registration
NAS1
NAS2
MN
xyz.com
MIPv6 signals in AAA messages
Connections between AAA infrastructures
Regular MIPv6 registration messages
Moving direction of mobile node

Fig. 5. Framework of TA^{4}ON

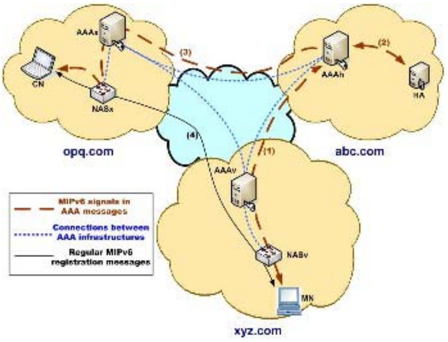

AAAx
CN
NASx
(3)
AAAh
(2)
HA
(4)
(1)
opq.com
abc.com
AAAv
NASv
MN
xyz.com
MIPv6 signals in AAA messages
Connections between AAA infrastructures
Regular MIPv6 registration messages

Fig. 6. Signals delivering and key materials distributing

3.3 Definition of TA^{4}ON Messages

MIPv6 signals are not only carried piggyback in some primitive AAA messages, but also delivered in some new types of message specially defined for MIPv6. Following is a list of primary types of new defined messages.

Type 1 (T1): handover registration. It is defined for delivering Binding Update (BU) and Binding Acknowledgment (BA) on handover registration, including home registration and correspondent registration. Note this type of message is not for the regular registration message sent between MN and HA/CN periodically.

Type 2 (T2): key material. It is defined for delivering security related data, such as secret key, lifetime and initial parameters and so on. Usually, key material is delivered to two peers, MN and HA or MN and CN, which need to initialize and protect their subsequent communication.

Type 3 (T3): dynamic home. It is defined for delivering Dynamic Home Agent Address Discovery (DHAAD) messages and Mobile Prefix Discovery (MPD) messages between MN and HA.

Type 4 (T4): network status. It is defined for delivering status information about involved network. AAA servers need to send probe packet to be aware of the network status. And if necessary, right current information about the network is delivered to some node which requires them.

Note, each type of message has options which can be used to differentiate subtypes.

Connections between any two AAA infrastructures are believed to be secure and stable, so we can use them to deliver some security sensitive messages. And any other signals of MIPv6 can be transmitted on this overlay network, too. Of course, if unnecessary, MIPv6 signals can be transmitted as usual.

3.4 Network Status Table

Every AAA server must maintain a Network Status Table (NST) to keep the current status of its domain and its neighbor domains. NST is used for interconnected AAA servers to find the way to other domain or sub-domain. In TA^{4}ON NST is a “virtual

Table 1. Network Status Table on AAAv

Sn	Domain Name	Next Peer
1	abc.com	aaav.abc.com
2	opq.com	aaax.opq.com
3	other.com	aaax.opq.com

topology table" for the overlay network. We believe that it must to some degree be a mirror of the real inter-network. So NST is necessary for TA^{4}ON to be "Topologically-Aware".

Table 1 is an sample of NST on AAAv in domain xyz.com. In Table 1, "Domain Name" is the destination domain. "Next Peer" is the valid way to that destination. Actually "Next Peer" is a AAA server. It is similar to the next hop in IP routing table. Entries in Table 1 must be symmetric, that is if AAAv has an entry for AAAx, then AAAx must has an entry for AAAv.

In addition, besides information in NST AAA server must know which MN is connecting to some NAS and which MN is registered at some HA. To do so, it must know information in its domain not only about NASs but also about HAs. AAA server must know which HA is in its domain and what its link prefix is. There is pre-established secure tunnel between AAA server and HA.

AAA servers send T4 messages with probe option to all its peers periodically to make sure that peer is alive and by the way get other status information about them. Probe T4 is sent not only to immediate peers but also to all other peers in NST because AAA servers must be "Topologically-Aware". AAA overlay network is setup by manual configuration because of administration and security, no dynamic routing algorithm is used. And so AAA overlay network is usually limited in a moderate scale.

4 Methodology

In this section, three typical usages of TA^{4}ON are described to show how TA^{4}ON enhances MIPv6.

4.1 Handover Signals Delivery

There are several solutions [9][22] for integrating MIPv6 home registration messages into AAA authentication messages in one round trip. This is supported by our proposal, too. See figure 6, following describes the procedure for delivering handover signals.

Step 1, when MN moves into a foreign network, it constructs authentication request message M$_1$ with its NAI and authentication data inside and with signal of MIPv6 home registration piggybacked.

$$M_1 = NAI_{mn}|Auth\_data_{mn}|MIPv6\_HR(HoA, CCoA) \tag{1}$$

Step 2, MN sends M$_1$ to NASv. NASv forwards M$_1$ to designated AAA server, AAAv. And then AAAv forwards M$_1$ to AAAh, the registry AAA server of MN (up direction of message flow 1 in figure 6). Before forwarding M_1, NASv and AAAv may store information and set soft states for M_1.

Step 3, on receiving M_1, AAAh authenticates MN with data inside. If successfully, AAAh exchanges T1 messages with HA to perform home registration in the name of MN (message flow 2 in figure 6).

Step 4, AAAh puts a successful BA into the authentication answer message and sends it back to MN (down direction of message flow 1 in figure 6).

Step 5, when MN receives a successfully authentication answer message from NASv, it knows that a) it is granted to access the network, b) home registration has been completed.

Above procedure has been used in other proposals [9][10][22]. However, in those proposals only HR was considered, CR was neglected. To speed up CR, besides operations described above, following additional operations are added.

At step 1, MN appends request message of CR to M_1, including HoA and CCoA of MN, NAI of CN. HoA and CCoA is for CN used to update binding list. NAIs is used for AAAh to identify CN's registry AAA server. At step 4, AAAh sends T1 message with HoA and CCoA to AAAx. AAAx checks whether this messages is valid. If so, AAAx forwards this message to NASx, NASx then forwards it to CN (flow 3 in figure 6). Usually at step 5, MN receives the answer from AAAh, CN almost at the same time receives this T1 message. Then in only one round trip authentication, HR and CR are all finished.

4.2 Key Material Distribution

In addition, in our proposal secret materials can be distributed over TA^{4}ON. There are at least two types of secret materials, one is between MN and HA, the other is between MN and CN.

Key materials between MN and HA can be used to setup dynamic SA between them. HA won't have to save SA for each MN registered at HA. MN won't save SA with its HA either. At MN only secret key with AAAh is remembered. When necessary, MN sends T2 request to AAAh, then AAAh generates a key material randomly [23] and sends T2 message over TA^{4}ON to MN and HA, respectively. Thus MN and HA can setup their SA with their secret material.

Key materials between MN and CN can be generated and distributed similarly. If AAAh receives a T2 request for this kind of key material, on sending key materials to MN, AAAh must identifies CN's registry AAA server and sends key materials to it. While MN and CN finally receive key material from their own registry AAA server respectively, they can work out a common Short-Term Static Key (STSK) according certain arrangement. After that, each time MN handover to new foreign link, RR is not necessary any more because they already have a short-term static key. CN can also send T2 request to its registry AAA server firstly, then key materials are generated and distributed by this AAA server.

STSK differs from the method described in subsection 4.1. They are used in different situation. The latter is used for CR on handover. STSK is used for BU sent periodically (flow 4 in figure 6).

To reduce the unnecessary workload of MN, CN and TA^{4}ON, T2 request for STSK can be sent as appendix of M_1 described in subsection 4.1 and it is feasible for MN and CN to use a STSK a few times until they believe that it should be replaced.

4.3 Dynamic Discovery of Mobile Prefix and Home Agent

In MIPv6 specification [1], it is allowed for home agent to change its address and even the topology of home network is allowed to change. The dynamic home agent discovery function could be used to learn addresses of home agents in the home network. But this is also an easy way for attacker to find target. Mobility prefix discovery is useful for MN to learn new topology of its home network. However, this must be done before the topology changes.

Here we use T3 messages to deal with this kind of problem. The prerequisite is that the address of AAAh will never change. Usually, it is truth in the real world.

When the address of home agent is changed or the topology of home network is changed, AAAh will be firstly informed in some manual way. So there are two cases for mobile node to learn about that. One is solicited mode. Mobile node send a T3 message with request option to AAAh, AAAh reply a T3 message with current information of home agents and home network to mobile node. The other is unsolicited mode. AAAh send T3 messages actively with new information about home agent and network to all involved mobile nodes as soon as anything changes in home network.

At the same time, secret materials are sent in T2 to related peers. Namely, if home agent is involved in above procedure, the new key material for IPSec SA_{mn-ha} is generated and sent by AAAh to mobile node and designated home agent respectively.

5 Analysis and Evaluation

5.1 Security Analysis

Above all, the security of TA^{4}ON is guaranteed by AAA protocol, so it is believed that TA^{4}ON is secure and stable. Communication between MN/CN and NASs is protected by key setup with the materials from AAA server during authentication procedure. Next, we discuss the security of MIPv6 when TA^{4}ON is used.

As far as HR and CR is concerned, signals are delivered over TA^{4}ON, so no security problem need worrying about. Communication between AAAh and HA is protected by preestablished secure tunnel. Communication between MN/CN and NASs is protected by key setup with the materials from AAA server during authentication procedure. So no vulnerability is introduced as long as TA^{4}ON is secure enough.

Key material distribution is secure, too. Most hops are in the TA^{4}ON. The final hop is between MN/CN and NASs. They are all under protection. Key material can be protected by additional encryption: registry AAA server encrypts key material by preshared SA before sending it to MN/CN; MN/CN decrypts it and generates STSK from it. Only MN and CN have the STSK, others, even NASs, have no idea about STSK. To determine how many times MN and CN use their STSK is not discussed in this paper.

MIPv6 signals between MN and CN are protected either by TA^{4}ON or by STSK, so RR procedure is canceled. Vulnerability introduced by RR is wiped out.

The security of DHAAD and MPD is also guaranteed by TA^{4}ON because all related signals are delivered on TA^{4}ON as well as key materials between peers.

5.2 Performance Analysis

TA^{4}ON is topologically-aware. Signals delivered on it can be thought to pass though a optimized route.

Similar with [22][10], HR over TA^{4}ON is performed in one trip. However, CR was not mentioned in [22][10]. CR over TA^{4}ON is finished in only one round trip, too. Usually, handover time can be derived out by following equation,

$$T = T_{IPv6} + T_{auth} + T_{hr} + T_{rr} + T_{cr} \tag{2}$$

where T is the total time of handover, T$_{IPv6}$ is time latency for IP address configuration and Duplicate Address Detection (DAD, if stateless auto-configuration is used). This is same in all solutions. T$_{auth}$ is the time latency for authentication. T$_{hr}$ is time latency for home registration. T$_{rr}$ is time latency for RR procedure and T$_{cr}$ is time latency for correspondent registration. To simple our discussion, T$_{IPv6}$ is ignored. Also comparing to delivery latency, processing is too small, so it is ignored, too. If we compute T with message round time, then we get following results.

$$T = T_{auth} + T_{hr} + T_{rr} + T_{cr} \tag{3}$$

In solution without optimization, T$_{auth}$=1, T$_{hr}$=1, T$_{rr}$=1.5 (path detouring HA is 1.5) and T$_{cr}$=1.

$$T_{no-opt} = 1 + 1 + 1.5 + 1 = 4.5 \tag{4}$$

In solution with half optimization, T$_{auth}$ + T$_{hr}$=1.1 (Latency between AAAh and HA is 0.1 because they in the same domain). Thus

$$T_{half-opt} = 1.1 + 1.5 + 1 = 3.6 \tag{5}$$

In our solution, T$_{auth}$ + T$_{hr}$ + T$_{cr}$ = 1.1, T$_{rr}$ = 0. Thus

$$T_{full-opt} = 1.1 + 0 = 1.1 \tag{6}$$

Comparing those results, we can say that our solution outperforms others.

If dynamic key is used between HA and MN, IKE is very time-consuming because there must be two phases and a few round trip in each phase. However, it is very easy and fast to implement dynamic key in our solution. AAAh can send dynamic key to HA and MN separately after a successful authentication. No extra count of round trip is introduced. Dynamic key distribution and authentication answer returning is finished simultaneously.

5.3 Additional Cost

Though security and performance of MIPv6 is improved, additional cost is added. Firstly, we have to take some cost to setup and maintain TA^{4}ON. This may be a demanding task. Secondly, protocols involved must be extended to support new operations. Finally, additional process and bandwidth occupation is inevitable.

However, TA^{4}ON maintenance is not conducted frequently. Protocol extension is done only once and only a little work is needed. Extra resource consumption is inevitable, but its benefit is noteworthy.

6 Conclusions and Future Work

A Topologically-Aware AAA Overlay Network (TA^{4}ON) is proposed for merging AAA mechanism and MIPv6. TA^{4}ON is aware of network topology and capable of security business. MIPv6's mobile signals and secret material can be delivered fast and securely through TA^{4}ON. Dynamic discovery of mobile prefix and home agent can be easily completed with the help of TA^{4}ON. So performance and security of MIPv6 handover are improved. Comparing to existing MIPv6-AAA methods, TA^{4}ON is not a partial but a total solution.

There still are a lot of work to do to make TA^{4}ON much more useful. TA^{4}ON may gain better performance if cooperates with HMIPv6. It may be helpful for MN and CN to optimize their routes, up stream and down stream separately. And TA^{4}ON may be used not only in MIPv6 environment but in any place where AAA mechanism is used.

References

1. D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", *IETF RFC3 775*, June 2004
2. R. Koodli, Ed., "Fast Handovers for Mobile IPv6", *IETF RFC 4068*, July 2005
3. H. Soliman, C. Castelluccia, K. El Malki, L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", *IETF RFC 4110*, August 2005
4. S. Kent, R. Atkinson, "IP Authentication Header (AH)", *IETF RFC 2402*, November 1998
5. S. Kent, R. Atkinson, "IP Encapsulating Security Payload (ESP)", *IETF RFC 2406*, November 1998
6. J. Arkko, V. Devarapalli, F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", *IETF RFC 3776*, June 2004
7. A. Patel, K. Leung, M. Khalil, H. Akhtar, K. Chowdhury, "Authentication Protocol for Mobile IPv6", *IETF-ID*, working in progress.
8. P. Calhoun, J. Loughney, E. Guttman, et al., "Diameter Base Protocol", *IETF RFC 3588*, June 2003
9. Franck Le, Basavaraj Patil, Charles E. Perkins, Stefano Faccin "Diameter Mobile IPv6 Application", *IETF-ID*, working in progress.
10. Kim C., Kim Y.S., et al., "Performance Improvement in Mobile IPv6 Using AAA and Fast Handoff", *Proc. of 2nd International Conference on Computer Science and its Applications (ICCSA'04)*, June 2004
11. Wang R.C., Chen R.Y., Chao H.C., AAA architecture for mobile IPv6 based on WLAN, *International Journal of Network Management*, Volume 14 , Issue 5, Pages: 305 C 313 ISSN:1099-1190, September 2004
12. R.I. C., Reen-Cheng, W., Han-Chieh, C., "Mobile IPv6 and AAA Architecture Based on WLAN", *Proc. of International Symposium on Applications and the Internet Workshops (SAINTW'04)*, January 2004
13. J. Kempf, J. Arkko, Nikander P., "Mobile IPv6 security", *ACM Wireless Personal Communications*, v29, n 3-4 SPEC.ISS., June, 2004, p 389-414

14. Charles E. Perkins, “Securing Mobile IPv6 Route Optimization Using a Static Shared Key”, *IETF-ID*, working in progress.
15. Ying Qiu, Jianying Zhou, Feng Bao, “Protecting All Traffic Channels in Mobile IPv6 Network”, *Proc. of IEEE Wirelsss Communications & Networking Conferance (WCNC’04)*, March 2004
16. S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, *IETF RFC 2460*, December 1998
17. Narten, T., Nordmark, E., Simpson, W. “Neighbor Discovery for IP Version 6 (IPv6)”, *IETF RFC 2461*, December 1998
18. A.D. Pramil, S. Antoine, A.H.Aghvami, TCP performance enhancement over mobile IPv6: innovative fragmentation avoidance and adaptive routing techniques,*IEE Proc.-Commun.*, Vol. 151, No. 4, August 2004
19. Kaufman, C., “Internet Key Exchange (IKEv2) Protocol”, *IETF-ID*, working in progress
20. Aboba, B. and M. Beadles, “The Network Access Identifier”, *IETF RFC2486*, January 1999
21. A. Patel, K. Leung,M. Khalil, *et al.*, “Mobile Node Identifier Option for MIPv6”, *IETF-ID*, working in progress
22. Jun Li, Xin-ming Ye, Jing-lin Shi, Miao Wang, ”Authenticated Stateful Auto-Configuration for Mobile IPv6 based on pre-IP Access Control”, *Proc. of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob’05)*August 2005
23. D. Eastlake, 3rd, J. Schiller,S. Crocker, Randomness Requirements for Security, *IETF RFC 4086*, June 2005
24. Gabriel Montenegro, Claude Castelluccia, “Crypto-based identifiers (CBIDs): Concepts and applications”, *ACM Transactions on Information and System Security (TISSEC)*, Volume 7 Issue 1, Pages: 97 - 127,ISSN:1094-9224, February 2004
25. T. Aura, “Cryptographically Generated Addresses (CGA)”, *IETF RFC 3972*, March 2005

QoS-Aware Multi-tier Location Managements for Integrated WLAN/UMTS Networks

Yun Won Chung

School of Electronic Engineering, Soongsil University,
511 Sangdo-Dong, Dongjak-Gu, Seoul 156-743, Korea
ywchung@ssu.ac.kr

Abstract. This paper addresses quality of service (QoS)-aware multi-tier location managements for integrated WLAN/UMTS networks. Single registration (SR) and multiple registration (MR) schemes are introduced and improved to support both voice and data services efficiently. The performance of the improved SR and MR schemes is analyzed based on mobility and traffic characteristics of mobile users. The results show that there is tradeoff between improved SR and MR schemes and thus, an appropriate scheme should be selected based on users' mobility and traffic characteristics. The analytical methodology developed in this paper can be extended to analyze the performance of general heterogeneous networks with multiple wireless access systems, which will be realized in fourth generation (4G) networks.

Keywords: Location management, multi-tier, registration, quality of service.

1 Introduction

Many wireless systems, such as cellular, wireless LAN (WLAN), mobile Ad hoc, sensor, digital video broadcasting (DVB), and satellite systems have been developed independently in order to meet the specific need of each system, such as cellular for high mobility and WLAN for high speed data service with low cost. Since these systems are also complimentary to each other in nature, the integration of these heterogeneous wireless systems is able to meet the ever increasing demand for anywhere, anytime, high speed data service.

The integration of heterogeneous wireless access systems based on common all-IP packet core is one of the most common visions for future mobile systems (e.g., 4G or beyond 3G) in mobile research community [1]. The 4G system supports global roaming to mobile users and satisfies their diverse service demands in all aspects, e.g., quality of service (QoS) and cost, via the best available access system.

In such a 4G system, global roaming should be supported always, even when mobile users move across heterogeneous wireless access systems, which makes the location management of 4G system very challenging. In location management, a mobile terminal (MT) notifies the network of its current registration area (RA) and it is stored in location register. Then, the location information is retrieved in call delivery process and paging is performed to find the cell of the MT within the RA.

In 4G system with various access systems, if there is no co-operative location management scheme, location updates are performed in all available systems independently,

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 307–318, 2006.

which results in high signalling load in access networks and high power consumption in MT due to multiple location updates sent by one MT. Reversely, paging is also simultaneously performed in all access systems redundantly and scarce radio resources are wasted.

In order to solve these problems, a new location management scheme is needed for heterogeneous network with multiple access systems. In this paper, we are primarily concerned with location management for integrated WLAN/UMTS networks as a preliminary study for location management for general heterogeneous network with multiple access systems because UMTS will be the main cellular system in a near future and WLAN has been widely deployed in hot spot areas.

There have been many studies on the location management for WLAN/UMTS networks. In 3GPP, feasibility on the interworking between UMTS and WLAN has been studied, where the interworking requirements and interworking scenarios are defined [2], [3]. In [4], a possible architecture for integrating UMTS and 802.11 WLAN is proposed, which allows an MT to maintain data connection through WLAN and voice connection through UMTS in parallel. In [5], mobility management procedures within and between UMTS and 802.11 WLAN for various integration scenarios are addressed and interaction between various networks elements and new procedures for seamless mobility are considered. In [6], dormant mode operation support for WLAN-UMTS roaming for dual-mode users is addressed.

These studies, however, only deal with interworking procedures and implementation aspects, but do not analyze the performance of the integrated WLAN/UMTS location management. A few studies have been conducted on the performance analysis of multi-tier location managements supporting voice service only [7] - [10]. In [7], single registration (SR) and multiple registration (MR) schemes with multi-tier home location register (MHLR) are proposed. In SR, an MT is allowed to register with MHLR on only one tier out of tiers available, where high tier corresponds to Advanced Mobile Phone System (AMPS) and low tier corresponds to Personal Access Communication System (PACS). Since low tier provides voice service with lower cost, priority is always given to low tier if both tiers are available. On the contrary, an MT is allowed to register with MHLR on multiple tiers at the same time in MR in order to remove frequent tier switchings in SR. From [7], it can be concluded that there is tradeoff between SR and MR schemes depending on the mobility and traffic characteristics. In [8] - [10], the performance of SR scheme and single-tier scheme has been analyzed in detail considering low-tier and high-tier registration time distributions.

These studies, however, only consider voice service but do not consider both voice and data services together. Thus, low tier always has higher priority than high tier, and if both low tier and high tier are available simultaneously, an MT is always registered at the low tier. In the integrated WLAN/UMTS networks, however, the type of service, i.e., voice service and data service, should be considered for tier selection because although low tier WLAN provides higher data rate services, UMTS should be used for voice service in order to meet the QoS requirement of realtime voice service. Thus, previous SR and MR schemes cannot be applied to systems directly considering both voice and data services, and appropriate extensions should be made.

Since current WLAN network is not appropriate for voice service with real-time service requirement, an MT should not change its network from UMTS to WLAN when the MT with voice service moves to WLAN coverage area. On the other hand, if the MT with data service moves to WLAN coverage area, it should change its network from UMTS to WLAN since WLAN supports higher data service. In this paper, we improve the SR and MR schemes proposed in [7] - [10] by considering both voice and data services together and analyze the performance tradeoff of these schemes.

This paper is organized as follows: Section 2 proposes the improved SR and MR schemes for integrated WLAN/UMTS networks. In Section 3, the performance of the improved SR and MR schemes is analyzed. Numerical examples are provided in terms of network signaling cost, radio signaling cost, and total signaling cost in Section 4. Finally, conclusions and further studies are presented in Section 5.

2 Improved Registration Schemes for Integrated WLAN/UMTS Networks

In this paper, we assume a simple WLAN/UMTS network area as shown in Fig. 1, where WLAN and UMTS networks are fully overlapped and a UMTS RA contains multiple WLAN hotspots. For simplicity, it is assumed that a WLAN hotspot consists of one WLAN RA. In [7] - [10], an MT is allowed to register with MHLR on only one tier in SR scheme and it is allowed to register with MHLR on multiple tiers at the same time in MR scheme. In SR scheme, there are two alternatives, i.e., SR1 and SR2 [7] - [10]. In SR1 scheme, the current RA of an MT is stored in MHLR. Suppose that the MT m moves between tiers as shown in Fig. 2, where it is in high tier RA H1 initially (1). In Fig. 2, high tier corresponds to UMTS and low tier corresponds to WLAN. Then, the location information is managed in MHLR as (m, H1). If the MT moves from high tier RA H1 to RA L1 of low tier (2), then there is registration request from the visitor location registration (VLR) of L1 to the MHLR and cancellation request is sent from the MHLR to the VLR of RA H1. In this case, (m, L1) is stored in MHLR. If the MT moves from RA L1 to RA H1 (3), registration from the VLR of RA H1 to MHLR and following cancellation from MHLR to the VLR of RA L1 are performed, and (m, H1) is stored in MHLR. Finally, if MT moves from RA H1 to RA L2 of low tier (4),

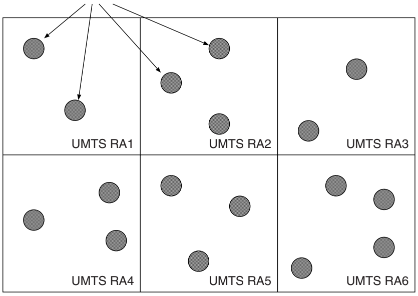


Fig. 1. A WLAN/UMTS network area

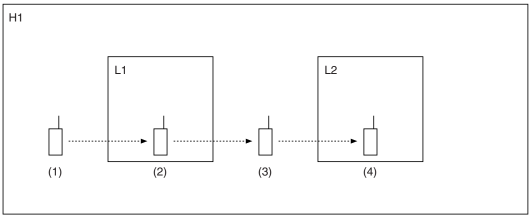


Fig. 2. Movement of MT between different tiers

registration from the VLR of RA L1 to the MHLR and following cancellation from MHLR to the VLR of RA H are performed, and (m, L2) is stored in MHLR.

In SR2 scheme, MHLR stores location information consisting of two location fields (one for high tier and one for low tier) and one bit to indicate the availability of tier [7] - [10]. Suppose that the MT m is in the high tier with RA H1 and the previous low tier RA is L1 (1). Then, location record (m,H;H1,L1) is stored in the MHLR. If the MT moves from RA H1 to low tier RA L1 (2), the VLR of RA L1 sends tier switching request to MHLR and the MHLR changes the tier bit to L, and the location record (m,L;H1,L1) is stored in the MHLR. In this case, there is no cancellation from MHLR to the VLR of RA H1. If the MT moves from RA L1 to RA H1 (3), only tier switching is performed and (m,H;H1,L1) is stored in the MHLR. Finally, if MT moves from RA H1 to RA L2 of low tier (4), registration and tier switching request are delivered from the VLR of RA L2 to MHLR and cancellation from MHLR to the VLR of L1 is performed. Location record (m,L;H1,L2) is stored in MHLR.

In MR scheme, MHLR stores most recently visited low tier RA and high tier RA information [7] - [10]. If the MT m is in H1 and the most recently visited low tier RA is L1 (1), (m,H1,L1) is stored in MHLR. If the MT moves to RA L1 (2), there is no registration and cancellation procedure since L1 is the same low tier RA as stored in the MHLR. If the MT moves from RA L1 to RA H1 of high tier (3), there is also no registration and cancellation procedure since H1 is the same high tier RA as stored in MHLR. Finally, if MT moves to RA L2 of low tier (4), VLR of RA L2 sends registration request to MHLR, cancellation request is sent to the VLR of RA L1 from MHLR, and (m,H1,L2) is stored in the MHLR. The location information of SR1, SR2, and MR schemes in MHLR is summarized in Table 1. In these SR1 and SR2 scheme, low tier has higher priority than high tier, and thus, if both low tier and high tier are available simultaneously, MT is always registered at the low tier. This is because although both low tier and high tier network support voice service, low cost is paid for using low tier.

Table 1. Location information of SR1, SR2, and MR schemes in MHLR

	(1)	(2)	(3)	(4)
SR1	(m,H1)	(m,L1)	(m,H1)	(m,L2)
SR2	(m,H;H1,L1)	(m,L;H1,L1)	(m,H;H1,L1)	(m,L,H1,L2)
MR	(m,H1,L1)	(m,H1,L1)	(m,H1,L1)	(m,H1,L2)

In this paper, however, we consider both voice service and data service together in WLAN/UMTS network. Since current WLAN does not satisfy QoS of realtime voice service, higher priority should be given to UMTS network for voice service, which is the major difference from the previous multi-tier location management schemes. On the contrary, WLAN should have higher priority for data service as in the previous schemes. Thus, SR1, SR2, and MR schemes should be improved appropriately in order to satisfy the QoS of both voice and data services in integrated WLAN/UMTS networks. Thus, we improved SR1, SR2, and MR, and they are denoted as ISR1, ISR2, and IMR, respectively.

In this paper, we consider the operation of ISR1, ISR2, and IMR for power on/off, registration, and voice/data call delivery procedures. When the MT is initially turned on, it performs power on registration. In ISR1, the registration request is sent through either WLAN or UMTS based on the network where the MT is located when the MT is turned on. If the MT is inside WLAN coverage area, it performs power on registration through WLAN. On the other hand, if the MT is outside WLAN coverage area, then power on registration is performed only through UMTS. In ISR2, power on registration is performed similarly as in ISR1. In IMR, the MT performs two separate power on registrations through both WLAN and UMTS when the MT is turned on inside WLAN coverage area. If the MT is outside WLAN coverage area, power on registration is performed only through UMTS.

When the MT is turned off, power off deregistration is performed. In ISR1, if the MT is inside WLAN when the MT is turned off, it only performs power off deregistration through WLAN network. On the other hand, in ISR2 and IMR, if the deregistration message is sent to MHLR via WLAN, additional deregistration message is sent from MHLR to UMTS. Likewise, if the deregistration message is sent to MHLR via UMTS, additional deregistration message is sent from MHLR to WLAN.

The number of network signaling and radio signaling messages for power on registration and power off deregistration is summarized in Table 2, where tier denotes at which tier an MT is located when the power on registration and power off deregistration events occur, and system denotes at which system either network or radio signaling messages occur.

Table 3 summarizes network signaling and radio signaling messages for registration by movement of MT, where P_{sw} is the probability that the WLAN network visited by MT is the same WLAN network that the MT visited most recently and direction means the direction of movement by the MT. In ISR1, there are always two signaling messages in both WLAN and UMTS for the movement in both directions. Note that there is no signaling message for a transition from low tier to high tier in IMR scheme. The number of messages can be obtained based on the operations of ISR1, ISR2, and IMR schemes.

Likewise, Table 4 summarizes the number of network signaling and radio signaling messages for voice/data call delivery, where N_{cell} denotes the number of cells in an RA in UMTS networks. We note that a WLAN RA is assumed to consist of single WLAN cell. In ISR1 and ISR2, voice/data calls should be delivered to the currently registered network irrespective the type of call. On the other hand, in IMR, voice call is always delivered through UMTS because WLAN is not appropriate for satisfying the QoS of voice service. For data call, since WLAN has higher priority than UMTS, it is firstly delivered through WLAN and if it fails, it is delivered to UMTS.

Table 2. Number of network signaling and radio signaling message for power on registration and power off deregistration

Scheme	Tier	System	Power on		Power off	
			Network	Radio	Network	Radio
ISR1	Low	WLAN	2	2	2	2
		UMTS	0	0	0	0
	High	WLAN	0	0	0	0
		UMTS	2	2	2	2
ISR2	Low	WLAN	2	2	2	2
		UMTS	0	0	2	0
	High	WLAN	0	0	2	0
		UMTS	2	2	2	2
IMR	Low	WLAN	2	2	2	2
		UMTS	2	2	2	0
	High	WLAN	0	0	2	2
		UMTS	2	2	2	0

Table 3. Number of network signaling and radio signaling message for registration by movement of MT

Scheme	Direction	System	Registration	
			Network	Radio
ISR1	High→Low	WLAN	2	2
		UMTS	2	0
	Low→High	WLAN	2	0
		UMTS	2	2
ISR2	High→Low	WLAN	2P$_{sw}$ + 4(1-P$_{sw}$)	2P$_{sw}$ + 2(1- P$_{sw}$)
		UMTS	0	0
	Low→High	WLAN	0	0
		UMTS	2	2
IMR	High→Low	WLAN	4(1- P$_{sw}$)	2(1- P$_{sw}$)
		UMTS	0	0
	Low→High	WLAN	0	0
		UMTS	0	0

3 Performance Analysis

For performance analysis, we make the following assumptions:

- Incoming voice call arrival and data call arrival at an MT occur according to a Poisson process with parameters λ_V and λ_D, respectively;
- WLAN residence time follows a general distribution with mean $1/\mu_w$;
- The interval from the time that an MT moves out of a WLAN coverage area to the time that the MT moves into next WLAN coverage area follows a general distribution with mean $1/\mu_u$.

Table 4. Number of network signaling and radio signaling message for voice/data call delivery

Scheme	Tier	System	Voice call		Data call	
			Network	Radio	Network	Radio
ISR1	Low	WLAN	2	2	2	2
		UMTS	0	0	0	0
	High	WLAN	0	0	0	0
		UMTS	2	N_{cell}+1	2	N_{cell}+1
ISR2	Low	WLAN	2	2	2	2
		UMTS	0	0	0	0
	High	WLAN	0	0	0	0
		UMTS	2	N_{cell}+1	2	N_{cell}+1
IMR	Low	WLAN	0	0	2	2
		UMTS	2	N_{cell}+1	0	0
	High	WLAN	0	0	0	0
		UMTS	2	N_{cell}+1	2	N_{cell}+1

In order to analyze the performance of ISR1, ISR2, and IMR, unit costs for network signaling load and radio signaling load are defined. The h_w^{reg}, h_w^{pag}, h_u^{reg}, and h_u^{pag} are defined as the unit network signaling cost for registration in WLAN, paging in WLAN, registration in UMTS, and paging in UMTS, respectively. The g_w^{reg}, g_w^{Vpag}, g_w^{Dpag}, g_u^{reg}, g_u^{Vpag}, and g_u^{Dpag} are defined as the unit radio signaling cost for registration in WLAN, paging for voice service in WLAN, paging for data service in WLAN, registration in UMTS, paging for voice service in UMTS, and paging for data service in UMTS, respectively.

Using the unit costs, network and radio signaling costs for power on registration for ISR1, ISR2, and IMR schemes are obtained based on Table 2 as follows:

$$\begin{aligned}
NC_{ISR1}^{on} &= 2P_w h_w^{reg} + 2P_u h_u^{reg}, && (1)\\
NC_{ISR2}^{on} &= 2P_w h_w^{reg} + 2P_u h_u^{reg}, && (2)\\
NC_{IMR}^{on} &= P_w(2h_w^{reg} + 2h_u^{reg}) + 2P_u h_u^{reg}, && (3)\\
RC_{ISR1}^{on} &= 2P_w g_w^{reg} + 2P_u g_u^{reg}, && (4)\\
RC_{ISR2}^{on} &= 2P_w g_w^{reg} + 2P_u g_u^{reg}, && (5)\\
RC_{IMR}^{on} &= P_w(2g_w^{reg} + 2g_u^{reg}) + 2P_u g_u^{reg}, && (6)
\end{aligned}$$

where P_w and P_u represent the probability that the MT is inside WLAN coverage and outside WLAN coverage when the MT is initially turned on and they are given as follows:

$$P_w = \frac{\mu_u}{\mu_w + \mu_u}, \qquad P_u = \frac{\mu_w}{\mu_w + \mu_u}. \tag{7}$$

Likewise, network and radio signaling costs for power off deregistration for ISR1, ISR2, and IMR schemes are obtained based on Table 2 as follows:

$$NC_{ISR1}^{off} = 2P_w h_w^{reg} + 2P_u h_u^{reg}, \tag{8}$$

$$NC_{ISR2}^{off} = 2P_w(2h_w^{reg}+2h_u^{reg})+P_u(2h_w^{reg}+2h_u^{reg}), \tag{9}$$

$$NC_{IMR}^{off} = P_w(2h_w^{reg} + 2h_u^{reg})+P_u(2h_w^{reg} + 2h_u^{reg}), \tag{10}$$

$$RC_{ISR1}^{off} = 2P_w g_w^{reg} + 2P_u g_u^{reg}, \tag{11}$$

$$RC_{ISR2}^{off} = 2P_w g_w^{reg} + 2P_u g_u^{reg}, \tag{12}$$

$$RC_{IMR}^{off} = 2P_w g_w^{reg} + 2P_u g_u^{reg}. \tag{13}$$

Network and radio signaling costs for registration for ISR1, ISR2, and IMR schemes are obtained based on Table 3 as follows:

$$NC_{ISR1}^{reg} = 2h_w^{reg} + 2h_u^{reg} + h_w^{reg} + 2h_u^{reg}, \tag{14}$$

$$NC_{ISR2}^{reg} = h_w^{reg}(2P_{sw} + 4(1 - P_{sw})) + 2h_u^{reg}, \tag{15}$$

$$NC_{IMR}^{reg} = 4h_w^{reg}(1 - P_{sw}), \tag{16}$$

$$RC_{ISR1}^{reg} = 2g_w^{reg} + 2g_u^{reg}, \tag{17}$$

$$RC_{ISR2}^{reg} = g_w^{reg}(2P_{sw} + 2(1 - P_{sw})) + 2g_u^{reg}, \tag{18}$$

$$RC_{IMR}^{reg} = 2g_w^{reg}(1 - P_{sw}). \tag{19}$$

Network and radio signaling costs for voice call delivery for ISR1, ISR2, and IMR schemes are obtained based on Table 4 as follows:

$$NC_{ISR1}^{Vpag} = 2P_w h_w^{pag} + 2P_u h_u^{pag}, \tag{20}$$

$$NC_{ISR2}^{Vpag} = 2P_w h_w^{pag} + 2P_u h_u^{pag}, \tag{21}$$

$$NC_{IMR}^{Vpag} = 2P_w h_u^{pag} + 2P_u h_u^{pag}, \tag{22}$$

$$RC_{ISR1}^{Vpag} = 2P_w g_w^{Vpag} + P_u(N_{cell} + 1)g_u^{Vpag}, \tag{23}$$

$$RC_{ISR2}^{Vpag} = 2P_w g_w^{Vpag} + P_u(N_{cell} + 1)g_u^{Vpag}, \tag{24}$$

$$RC_{IMR}^{Vpag} = P_w(N_{cell} + 1)g_u^{Vpag} + P_u(N_{cell} + 1)g_u^{Vpag}. \tag{25}$$

Finally, network and radio signaling costs for data call delivery for ISR1, ISR2, and IMR schemes are obtained based on Table 4 as follows:

$$NC_{ISR1}^{Dpag} = 2P_w h_w^{pag} + 2P_u h_u^{pag}, \tag{26}$$

$$NC_{ISR2}^{Dpag} = 2P_w h_w^{pag} + 2P_u h_u^{pag}, \tag{27}$$

$$NC_{IMR}^{Dpag} = 2P_w h_u^{pag} + 2P_u h_u^{pag}, \tag{28}$$

$$RC_{ISR1}^{Dpag} = 2P_w g_w^{Dpag} + P_u(N_{cell} + 1)g_u^{Dpag}, \tag{29}$$

$$RC_{ISR2}^{Dpag} = 2P_w g_w^{Dpag} + P_u(N_{cell} + 1)g_u^{Dpag}, \tag{30}$$

$$RC_{IMR}^{Dpag} = 2P_w g_w^{Dpag}+P_u(g_w^{Dpag} + (N_{cell} + 1)g_u^{Dpag}). \tag{31}$$

Based on the above results, total expected network and radio signaling costs for each scheme are derived as follows:

$$\begin{aligned} NC_i &= NC_i^{on} + NC_i^{off} + NC_i^{reg} N_{cycle} \\ &+ (NC_i^{Vpag} \lambda_V (1/\mu_w + 1/\mu_u) \\ &+ NC_i^{Dpag} \lambda_D (1/\mu_w + 1/\mu_u)) N_{cycle}, \end{aligned} \tag{32}$$

$$\begin{aligned} RC_i &= RC_i^{on} + RC_i^{off} + RC_i^{reg} N_{cycle} \\ &+ (RC_i^{Vpag} \lambda_V (1/\mu_w + 1/\mu_u) \\ &+ RC_i^{Dpag} \lambda_D (1/\mu_w + 1/\mu_u)) N_{cycle}, \end{aligned} \tag{33}$$

where i denotes each location registration scheme, i.e., ISR1, ISR2, and IMR, and N_{cycle} is the total expected number of the movement into or out of WLAN coverage area during power on period, which is obtained by $\frac{1/\mu_a}{1/\mu_u + 1/\mu_w}$. Finally, total signaling cost for each scheme is defined as follows:

$$C_i = w_{net} NC_i + w_{rad} RC_i, \tag{34}$$

where w_{net} and w_{rad} are weighting factors for network signaling cost and radio signaling cost, respectively since these two costs are not directly comparable.

4 Numerical Examples

In this section, numerical examples are given in order to investigate the tradeoff among the ISR1, ISR2, and IMR. Default parameter values for network and radio signaling cost, and mobility and traffic characteristics are summarized in Tables 5 and 6, respectively. In Table 5, we give higher values for unit radio signaling cost for registration than that for paging because it is widely accepted that more data bytes are needed for registration and thus, consumes more radio bandwidth.

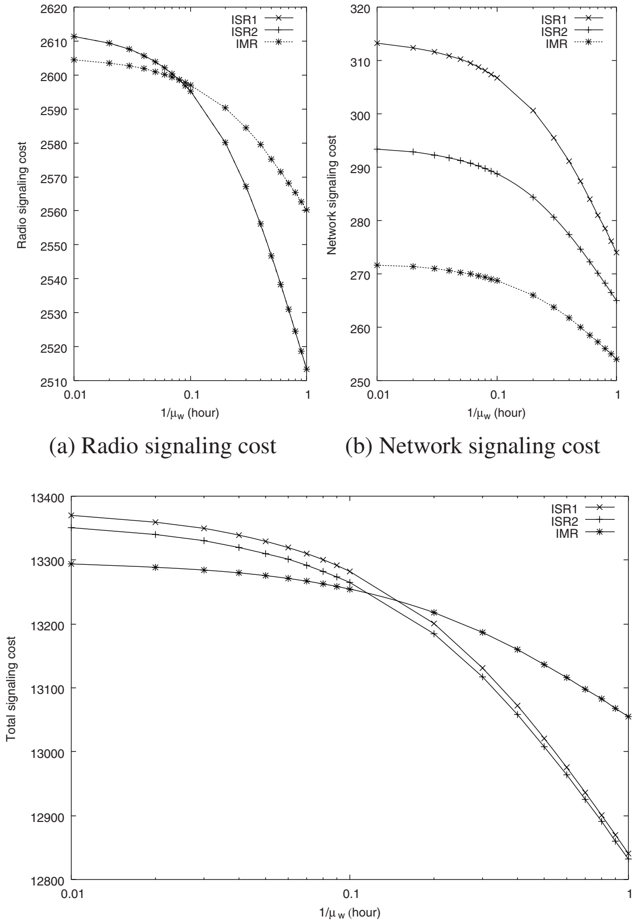
Figs. 3 (a), (b), and (c) show radio, network, and total signaling cost for varying the values of $1/\mu_w$. As can be seen in Fig. 3 (a), ISR1 and ISR2 have the same radio signaling cost, as expected. For small values of $1/\mu_w$, i.e., frequent movements out of WLAN coverage area, IMR performs better than ISR1 and ISR2 due to less registration

Table 5. Default parameter values for unit network and radio signaling costs

h_w^{reg}	h_w^{pag}	h_u^{reg}	h_u^{pag}	g_w^{reg}	g_w^{Vpag}	g_w^{Dpag}	g_u^{reg}	g_u^{Vpag}	g_u^{Dpag}
1	1	1	1	5	1	1	5	1	1

Table 6. Default parameter values for mobility and traffic characteristics

λ_V	λ_D	μ_a	μ_w	μ_u	P_{sw}	N_{cycle}	N_{cell}	w_{net}	w_{rad}
1.5	10	0.1	10	1	0.1	20	20	1	5



(a) Radio signaling cost (b) Network signaling cost

(c)Total signaling cost

Fig. 3. Signaling cost for varying $1/\mu_w$

signaling cost. For large values of $1/\mu_w$, on the other hand, ISR1 and ISR2 perform better than IMR due to less paging signaling cost. Since the number of tier switching is the smallest in IMR, network signaling cost of IMR is the smallest. On the other hand, the network signaling cost of ISR1 is the largest due to the largest number of tier switching. It can be seen that ISR2 performs better than ISR1 for the considered parameter sets and there is tradeoff between IMR and others.

Fig. 4 shows total signaling for varying the values of P_{sw}. As can be expected, the total signaling cost of IMR decreases as the value of P_{sw} increases because if P_{sw} is high, the probability of tier switching in IMR is small. We note that the total signaling cost of ISR1 does not vary for the change of P_{sw} because new registration at new tier and new deregistration at old tier always occur due to the movement between low tiers and high tiers by MT. There is also tradeoff among the three schemes.

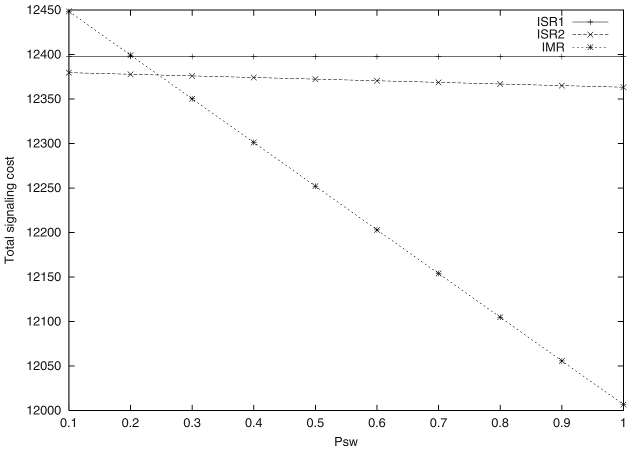


Fig. 4. Total signaling cost for varying P_{sw}

5 Conclusions and Further Studies

In this paper, conventional SR and MR schemes are improved appropriately in order to accommodate both voice and data services appropriately in integrated WLAN/UMTS networks. The number of network signaling and radio signaling messages for power on registration, power off deregistration, registration by movement of MT, and voice/data call delivery is obtained. Then, the performance of ISR1, ISR2, and IMR schemes is analyzed based on mobility and traffic characteristics of mobile users in terms of radio, network, and total signaling costs. The numerical results investigate the performance variation of the proposed schemes for varying the characteristics parameter. It is concluded that there is tradeoff among the three schemes and thus, an appropriate scheme should be selected based on mobility and traffic characteristics. The analytical methodology developed in this paper can be extended to analyze the performance of general heterogeneous networks with multiple wireless access systems, which will be realized in 4G networks. As further studies, we are doing research on the following topics:

- More realistic modelling of integrated WLAN/UMTS network residence time distributions;
- More accurate analysis of network, radio, and total costs based on more detailed modelling of mobility and traffic characteristics of mobile users;
- Dynamic selection of multi-tier location management schemes based on mobile user's varying mobility and traffic characteristics.

These further studies can help to solve the location management problem in 4G heterogeneous networks with multiple wireless access systems such as GSM, UMTS, WLAN, Satellite, and DVB, with the aid of the analytical methodology developed in this paper.

Acknowledgement

This work was supported by the Soongsil University Research Fund.

References

1. I. F. Akyildiz, J. Xie, and S. Mohanty, “A survey of mobility management in next-generation all-IP-based wireless systems,” *IEEE Wireless Communications,* pp. 16-28, Aug. 2004.
2. 3GPP TR 22.934 V6.2.0, “Feasibility study on 3GPP system to wireless local area network (WLAN) interworking,” Sep. 2003.
3. 3GPP TR 23.234, “3GPP system to wireless local area network (WLAN) interworking; system description,” Jan. 2004.
4. M. Jaseemuddin, “Architecture for integrating UMTS and 802.11 WLAN networks,” *IEEE International Symposium on Computers and Communication,* pp. 2003.
5. V. K. Varma, S. Ramesh, K. D. Wong, M. Barton, G. Hayward, and J. A. Friedhoffer, “Mobility management in integrated UMTS/WLAN networks,” *IEEE International Conference on Communications,* pp. 1048-1053, 2003.
6. B. Sarikaya and T. Ozugur, “Dormant mode operation support for roaming from WLAN to UMTS,” *IEEE International Conference on Communications,* pp. 1038-1042, 2003.
7. Y. B. Lin and I. Chlamtac, “Heterogeneous personal communications services: integration of PCS systems,” *IEEE Communications Magazine,* vol. 3, no. 9, pp.106-112, Sep. 1996.
8. Y. B. Lin, “A comparison study of the two-tier and the single-tier personal communications service systems,” *ACM/Baltzer Mobile Networks and Applications,* vol. 1, no. 1, pp. 29-38, 1996.
9. Y. Fang, “Registration traffic and service availability for two-tier wireless networks,” *IEEE Wireless Communications and Networking Conference,* pp. 1090-1095, 2000.
10. Y. Fang and Y.B. Lin, “Mobility management and signaling traffic analysis for multi-tier wireless mobile metworks,” *IEEE Transactions on Vehicular Technology,* vol. 54, no. 5, pp. 1843-1853, Sept. 2005.

Leveraging Buffering Delay Estimation for Geolocation of Internet Hosts

Bamba Gueye[1], Steve Uhlig[2,⋆], Artur Ziviani[3], and Serge Fdida[1]

[1] Université Pierre et Marie Curie,
Laboratoire d'Informatique de Paris 6 (LIP6)
{gueye, fdida}@rp.lip6.fr
[2] Université Catholique de Louvain,
Department of Computing Science and Engineering
suh@info.ucl.ac.be
[3] National Laboratory for Scientific Computing (LNCC)
ziviani@lncc.br

Abstract. Geolocation techniques aim at determining the geographic location of an Internet host based on its IP address. Currently, measurement-based geolocation techniques disregard the buffering delays that may be introduced at each hop along the path taken by probe packets. To fill this gap, we propose the *GeoBuD* (**Geo**location using **Bu**ffering **D**elay estimation) approach. Although the network delay and the geographic distance between two Internet hosts have been shown to be related to some extent, leveraging buffering delay estimation at each hop for geolocation purposes is challenging for two reasons. First, correctly estimating the buffering delay at intermediate hops along a traceroute path for geolocation purposes depends on the accurate estimation of the geolocation of the intermediate routers. Second, even given an *a priori* knowledge of the location of the routers, estimating the buffering delays is difficult due to the coarse-grained information provided by delay measurements. Relying on traceroute measurements, we show that leveraging buffering delay estimation improves accuracy in the measurement-based geolocation of Internet hosts as well as the confidence that the geolocation service associates to each estimation.

Keywords: geolocation, buffering delay estimation, traceroute, multilateration.

1 Introduction

Geographically locating an Internet host from its IP address enables a diversified class of location-aware applications [1, 2, 3]. Examples of such applications comprise targeted advertising on web pages, displaying local events and regional weather, automatic selection of a language to first display the content of web pages, restricted content delivery following regional policies, authorization of transactions only when performed from pre-established locations, or locating pedo-criminality. Each application may have a different requirement on the resolution of the location estimation. Nevertheless, as IP

[⋆] Steve Uhlig is Postdoctoral fellow of the Belgian National Fund for Scientific Research (F.N.R.S).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 319–330, 2006.

addresses are in general allocated in an arbitrary fashion, there is no inherent relation between an IP address and the physical location of the corresponding physical interface. Therefore, inferring the geographic location of Internet hosts is a challenging problem.

Previous work on measurement-based geographic location of Internet hosts [4, 5, 6] relies on delay measurements between *landmarks*, *i.e.* hosts with well-known geographic location, to provide the position of a target host. In GeoPing [4], the positions of landmarks are used as the possible location estimates for a given target host. This leads to a discrete space of answers that may limit location accuracy because of the system's dependence on the number and placement of landmarks [5]. The *Constraint-Based Geolocation* (CBG) approach proposed by Gueye *et al.* [6] transforms delay measurements into distance constraints and then uses *multilateration* to estimate the geographic location of a given target host. Multilateration refers to the process of estimating a position using a sufficient number of distances to some fixed points, thus establishing a continuous space of answers instead of a discrete one. This multilateration with distance constraints provides an overestimation of the distance from each landmark to the target host to be located, thus determining a region that hopefully encloses the location of the target host. The centroid of this region is the location estimation provided by CBG. Further, the area size of this region is a confidence measure CBG associates with each given location estimation; the smaller the region, the more confident the system is in the provided estimation. Although showing relatively accurate results in most cases, these measurement-based approaches may have their accuracy disturbed by many sources of distortion that affect delay measurements. For example, delay distortion may be introduced by the circuitous Internet paths that tend to unnecessarily inflate the end-to-end delay [7, 8, 9]. Another source of distortion is the unpredictable buffering delay that packets face in queues at the intermediate routers along the end-to-end path. For an accurate geolocation of Internet hosts based on delay measurements, it is crucial to estimate and remove as much of the additional delay as possible.

This paper investigates the distortion introduced in delay measurements due to buffering delays and possible counter-measures to it in order to improve geolocation techniques. We present GeoBuD, a novel way of geolocating Internet hosts. We rely on traceroute measurements to estimate the buffering delay introduced along the path from each landmark to the target. Based on traceroute information about the successive RTTs at each intermediate hop, we estimate the buffering delay introduced by each of these hops. Our results show that the estimation of buffering delays introduced along the path allows the improvement of the geolocation estimation given by CBG. This is so because the additional delay distortions caused by buffering delay are removed from the overestimations of distance constraints that define the region enclosing the target host in CBG, thus allowing tighter overestimations that result in a smaller region. Smaller regions that still enclose the target host provide more accurate location estimation in CBG.

The remainder of the paper is structured as follows. Section 2 discusses the related work. Section 3 describes the CBG approach to estimate the geographic location of a given target host. Section 4 explains our methodology for performing the traceroutes and estimating the buffering delays along the traceroute measurements. Section 5 compares the results of GeoBuD and those of the CBG approach. Finally, Section 6 concludes our paper and discusses future work.

2 Related work

A DNS-based approach to provide a geographic location service of Internet hosts is proposed in RFC 1876 [10]. Nevertheless, the adoption of the DNS-based approach has been limited since it requires changes in the DNS records and administrators have little motivation to register new location records. Tools such as IP2LL [11] and NetGeo [12] query Whois databases in order to obtain the location information recorded therein to infer the geographic location of a host. This information, however, may be inaccurate or stale. Moreover, if a large and geographically dispersed block of IP addresses [13] is allocated to a single entity, the Whois databases may contain just a single entry for the entire block.

There are also some geolocation services based on an exhaustive tabulation between IP addresses ranges and their corresponding locations. Examples of such services are GeoURL [14], the Net World Map project [15], and several commercial tools [1, 2, 3].

Padmanabhan and Subramanian [4] investigate three different techniques to infer the geographic location of an Internet host:

- The first technique infers the location of a host based on the DNS name of the host or another nearby node. This technique is the basis of GeoTrack [4], VisualRoute [16], GTrace [17], and the SarangWorld Traceroute project [18]. Quite often network operators assign names to routers that have some geographic meaning, presumably for administrative convenience. Nevertheless, not all names contain an indication of location. Since there is no standard, operators commonly develop their own rules for naming their routers even if the names are geographically meaningful. Therefore, the parsing rules to recognize a location from a node name must be specific to each operator. The creation and management of such rules is a challenging task as there is no standard to follow.
- The second technique splits the IP address space into clusters such that all hosts with an IP address within a cluster are likely to be co-located. Knowing the location of some hosts in the cluster and assuming they are in agreement, the technique infers the location of the entire cluster. An example of such a technique is GeoCluster [4]. This technique, however, relies on information that is partial and possibly inaccurate. The information is partial because it comprises location information for a relatively small subset of the IP address space. Moreover, such information may be inaccurate because the databases rely on data provided by users, which may be unreliable.
- The third technique, GeoPing [4], is based on exploiting a possible correlation between geographic distance and network delay. The location estimation of a host is based on the assumption that hosts with similar network delays to some fixed probe machines tend to be located near each other. This assumption is similar to the one exploited by wireless positioning systems such as RADAR [19] concerning the relationship between signal strength and distance. Therefore, given a set of landmarks with a well-known geographic location, the location estimation for a target host is the location of the landmark presenting the most similar delay pattern to the one observed for the target host. In GeoPing, the number of possible location estimates is limited to the number of adopted landmarks, characterizing a discrete space of

answers. As a consequence, the accuracy of this discrete space system is directly related to the number and placement of the adopted landmarks [5].

To overcome the limitation of using a discrete space of answers, the Constraint-Based Geolocation [6] approach uses multilateration to yield a continuous space of answers. In the next section, we provide a brief background on how the CBG methodology operates as the goal in this paper is to investigate the impact that the leveraging of buffering delay can have on the geolocation of Internet hosts based on multilateration.

3 Background on the CBG Approach

In this section, we present a brief background on how CBG provides geolocation estimation for target hosts based on delay measurements.

3.1 Multilateration with Geographic Distance Constraints

The physical position of a given point can be estimated using a sufficient number of distances or angle measurements to some fixed points whose positions are known. When dealing with distances, this process is called multilateration.

Consider a set $\mathcal{L} = \{L_1, L_2, \ldots, L_K\}$ of K landmarks. Landmarks are reference hosts with a well-known geographic location. For the location of Internet hosts using multilateration, CBG [6] tackles the problem of estimating the geographic distance from these landmarks towards the target host to be located, given the delay measurements from the landmarks. From a measurement viewpoint, the end-to-end delay over a fixed path can be split into two components: a deterministic (or fixed) delay and a stochastic delay [20]. The deterministic delay is composed by the minimum processing time at each router, the transmission delay, and the propagation delay. This deterministic delay is fixed for any given path. The stochastic delay comprises the queuing delay at the intermediate routers and the variable processing time at each router that exceeds the minimum processing time. Besides the stochastic delay, the conversion from delay measurements to geographic distance is also distorted by other sources as well, such as circuitous routing and the presence of redundant data. Anyway, it should be noted that no matter the source of distortion, this delay distortion is always additive with respect to the minimum delay of an idealized direct great-circle path.

Figure 1 illustrates the multilateration in CBG using the set of landmarks $\mathcal{L} = \{L_1, L_2, L_3\}$ in the presence of some additive distance distortion due to imperfect measurements. Each landmark L_i intends to infer its geographic distance constraint to a target host τ with unknown geographic location. Nevertheless, the inferred geographic distance constraint is actually given by $\hat{g}_{i\tau} = g_{i\tau} + \gamma_{i\tau}$, *i.e.* the real geographic distance $g_{i\tau}$ plus an additive geographic distance distortion represented by $\gamma_{i\tau}$. This purely additive distance distortion $\gamma_{i\tau}$ results from the possible presence of some additive delay distortion. As a consequence of having additive distance distortion, the location estimation of the target host τ should lie somewhere within the gray area (*cf.* Figure 1) that corresponds to the intersection of the overestimated geographic distance constraints from the landmarks to the target host.

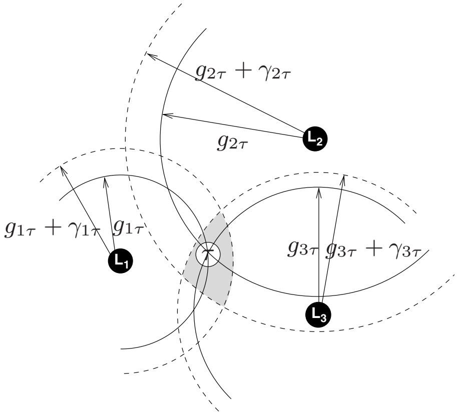

$g_{2\tau} + \gamma_{2\tau}$
$g_{2\tau}$
L2
$g_{1\tau} + \gamma_{1\tau}$
$g_{1\tau}$
L1
τ
$g_{3\tau}$
$g_{3\tau} + \gamma_{3\tau}$
L3

Fig. 1. Multilateration with geographic distance constraints

3.2 From Delay Measurements to Distance Constraints

Recent work [21, 4, 22] has investigated the correlation between geographic distance and network delay. Figure 2 provides an example of the relation between the distance and the delay for one of the landmarks we used in our measurements towards the remaining landmarks of our dataset (further details on the experimental data used are found in Section 5). The *bestline* shown in Figure 2 for a given landmark L_i is defined as the line that is closest to, but below all data points (x, y), where x expresses the actual great-circle geographic distance between this given landmark and all the other landmarks in the set, while y represents the measured RTT between the same pairs. The equation of the bestline is defined as

$$y = m_i x + b_i. \tag{1}$$

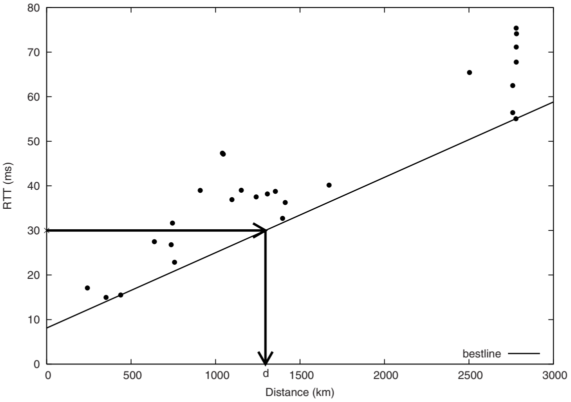


Fig. 2. Sample scatter plot of geographic distance and network delay

It should be noted that each landmark finds its slope m_i and its positive intercept b_i based only on delay measurements between the available landmarks. For further details about the computation of b_i and m_i, we refer the reader to [6]. The presence of a positive intercept b_i in the bestline reflects the presence of some localized delay. Each landmark uses its own bestline to convert the delay measurement towards the target host into a geographic distance constraint. A delay measurement from the considered landmark of Figure 2 towards a particular target host τ is transformed into a distance constraint by projecting the measured delay on the distance axis using the computed bestline of this landmark. For example, if the measured delay is 30 ms, the distance constraint is d, as illustrated by the thick arrow in Figure 2. This estimated geographic distance constraint $\hat{g}_{i\tau}$ between a landmark L_i and a target host τ is derived from the delay $d_{i\tau}$ using the bestline of the landmark as follows:

$$\hat{g}_{i\tau} = \frac{d_{i\tau} - b_i}{m_i}. \tag{2}$$

Each landmark L_i localizes a given destination τ inside a circle whose radius is the obtained distance constraint $\hat{g}_{i\tau}$. The region formed by the intersection of all these circles from the set of landmarks is called in CBG the *confidence region*. CBG provides the centroid of this confidence region as the location estimation for the target host.

4 Buffering Delay Estimation Via Traceroutes

CBG builds its distance constraints on a per-landmark basis. In contrast, based on a per-destination, GeoBuD transforms delay measurements into distance constraints. In practice, paths for different destinations may suffer from different distortions. To take that into account, we first replace the linear model of CBG by decomposing it in a per-hop basis. So in the GeoBuD approach, for each landmark L_i and each target host τ, we model the delay $y_{i\tau}$ as

$$y_{i\tau} = m_i x_{i\tau} + b_{i\tau}, \tag{3}$$

where m_i represents the propagation speed of data along the path computed only between the landmarks, $x_{i\tau}$ represents the geographic distance constraint between landmark L_i and destination τ, and $b_{i\tau}$ represents the total buffering delay along the path from L_i to a target host τ. In our measurements, the value of m_i actually represents 2 times the propagation speed of light in fiber, as m_i captures both the signal propagation aspect and the fact that the delay on which we are relying is a RTT hence contains both the forward and the return path. To estimate the total buffering delay $b_{i\tau}$, we estimate the buffering delay at each hop along the path from L_i to target τ based on traceroute between these nodes. The output of the traceroute measurements hopefully provides the different intermediate nodes that compose the path, as well as the delay between each pair of consecutive intermediate router.

For example, suppose that we perform a traceroute from landmark L_i towards some target host τ. The traceroute is composed of n intermediate hops, the last hop being the one that arrives at the target node. For each hop k of the traceroute that answers with an ICMP message `TIME exceeded`, we have an RTT measurement. If by any reason an intermediate router along the traceroute path does not answer with the ICMP message

`TIME exceeded`, we disconsider this hop as we lack a delay measurement for this particular node. To estimate $b_{i\tau}$ in Equation (3), we actually estimate its components b_k along the traceroute path using

$$\Delta RTT_{k+1} = RTT_{k+1} - RTT_k = m_i \times dist(k, k+1) + b_{k+1}, \tag{4}$$

where k represents the k^{th} intermediate router on the traceroute path for which we were able to have a delay measurement and geographical location. The term RTT_k denotes the minimum RTT value out of the 3 RTTs measurements obtained for a given hop[1] k and *dist(k,k+1)* represents the geographic distance between nodes k and $k+1$. Note that m_i is the same as the one in Equation (1). The sum of the $dist(k, k+1)$ for each k from 0 to $n-1$ gives the estimation of the geographic path length followed by the traceroute. Thus, we estimate the buffering delay b_k at each hop k in a straightforward way from Equation (4) as

$$b_k = \Delta RTT_k - m_i \times dist(k-1, k). \tag{5}$$

It is clear from Equation (5) that we need to estimate the geographic distance between each pair of consecutive intermediate routers along the traceroute path in order to be able to estimate b_k. This implies knowing the geographic location of these routers. It is unlikely to have an *a priori* knowledge of the geographic location of all possible intermediate routers along a traceroute path. This would actually amount to being able to geolocate any Internet node, *i.e.* the actual intent of the geolocation service under investigation. Therefore, the estimation of buffering delay along the path demands successive use of the geolocation service on each node identified along the traceroute path until reaching the target host.

5 GeoBuD Evaluation

In order to estimate the buffering delay b_k and the $dist(k, k+1)$ at each hop k along a path between each landmark and each target host τ, we have considered two datasets: First, we considered nodes located in the U.S. We have used 29 PlanetLab nodes [24] as landmarks and 87 AMP nodes [25] as targets. The dataset we consider is composed by traceroute measurements performed on October 17^{th} 2005 from our landmarks towards targets hosts. For the second dataset we performed on November 21^{st} 2005 traceroutes from 27 PlanetLab nodes located in Western Europe towards 57 RIPE nodes [26], also located in Western Europe. In CBG methodology, landmarks perform ping measurements towards a given target host to locate it. Traceroute measurements from the same landmarks towards the same targets were performed simultaneously with ping measurements, in order to have similar network conditions for both CBG and traceroute measurements.

In our traceroute experiments, for the U.S. dataset, we have been able to geolocate 1153 distinct intermediate routers excluding the AMP hosts, out of a total of 1408 traversed routers, thus leading to geolocating 82% of the intermediate routers. In the W.E

[1] Each step of a traceroute consists in sending 3 consecutive UDP packets towards the destination using an increasing TTL value. In our measurements we rely on native traceroutes [23].

dataset we have located 1235 routers among the 1328 routers that we have encountered. So 93% of W.E. routers are located. It should be noted that most of undiscovered routers are typically located in the vicinity of the source or the destination, so the resulting error in the traceroute path lentgh estimation is due to be small. For each of these located routers, we relied on the CBG-based GeoLIM project [27] to find out their geographic location. We cross-checked the results obtained with the GeoLIM project with *rockettrace* provided by the scriptroute tool suite [28]. In addition, some hops along the traceroutes underwent congestion by the time the measurements are carried. These traffic conditions, however, may not last for the whole time of the traceroute measurements. In such a case, only a few intermediate hops along the traceroute path exhibit a very large RTT value. If any of the intermediate hops along a traceroute exhibits a RTT value larger than any RTT of its succeeding hops along the traceroute path, we disconsider this particular hop. If we were to take such inflated RTTs into account we would overestimate the buffering delay for that hop.

After performing the geolocation of the intermediate routers, we compute the set of b_k values along each of different traceroute paths for the located intermediate nodes using Equation (4). In some cases, the estimated b_k are negative, in which case they were not considered. We had 21% of negative b_k's corresponding to 4043 among 19172 b_k's that we have computed for AMP hosts. For RIPE hosts the percentage of negative b_k's is 14% for 11908 b_k's found. Most cases where the b_k were negative correspond to situations where $\varDelta RTT_{k+1}$ is very small or negative. This is due to variations in the network conditions along the path of the traceroute during the experiments. Hence, to consider a particular b_k, we require that $\varDelta RTT_{k+1} > 0$.

For each landmark L_i and target τ, we have then a corresponding $b_{i\tau} = \sum_{k=1}^{n-1} b_k$, where n denotes the number of intermediate hops along the traceroute path from landmark L_i and target τ. To transform delay measurements into distance constraints, we can use the following equation derived from Equation (3):

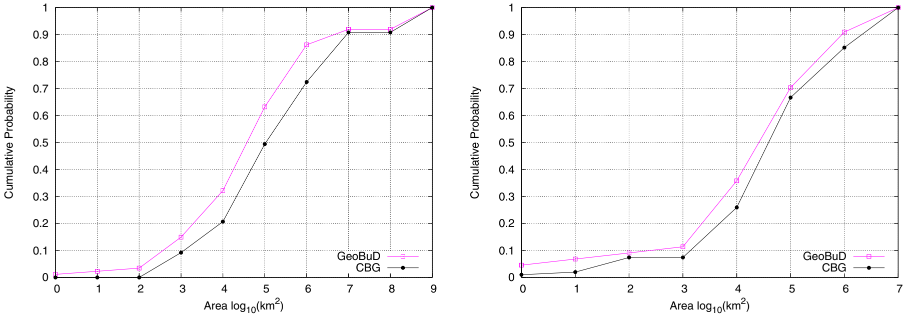
$$x_{i\tau} = \frac{y_{i\tau} - b_{i\tau}}{m_i}. \tag{6}$$

GeoBuD uses the distance constraints given by Equation (6) to localize a given target host. The distance constraints obtained by GeoBuD are expected to be tighter than those provided by the CBG method. Using these new tighter distance constraints, in spite of the number of negative b_k's, the confidence region shrinks, thus increasing both the accuracy of the location estimation and the system's confidence on these location estimation as shown in Section 5.1.

5.1 Shrinking the Confidence Region

Figure 3 compares the cumulative probability distributions of the confidence region of GeoBuD and the CBG approach. On the x-axis, we have the surface area of the confidence region for different target hosts. On the y-axis, we show the probability that the location estimation for the target hosts have a confidence region smaller than x.

One can observe on Figure 3 the improvement due to buffering delay estimation for areas smaller than 10^7 km^2. With CBG, 72% of the target hosts located in the U.S. have a confidence region smaller than 10^6 km^2. For GeoBuD and the same confidence region surface, we have about 86% of the target hosts. With CBG, 49% of the target hosts have a confidence region smaller than 10^5 km^2, whereas for GeoBuD 63% of the target hosts are within such a confidence region. For hosts located in W.E., GeoBuD localizes 10% of the target hosts with a confidence region inferior to 10^2 km^2. For reference, a surface area of 10^5 km^2 is slightly larger than Portugal or the U.S. state of Indiana.



(a) U.S. dataset

(b) Western Europe dataset

Fig. 3. Confidence regions provided by GeoBuD and CBG in km^2

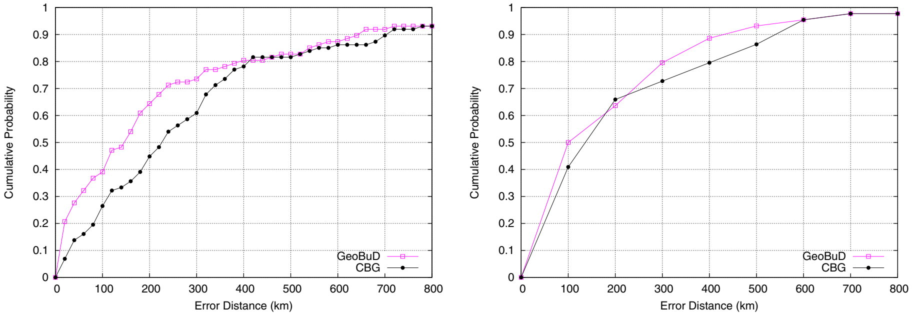
5.2 Location Estimation Error

We might expect that reducing the surface area of the confidence region by estimating the buffering delay would also reduce the error observed in location estimation. In Figure 4, we show the cumulative probability of the error observed in the obtained location estimation. The estimation error for a given target host is the difference between its actual geographic location and its location estimate. The performance gap between GeoBuD and CBG is larger in the U.S. dataset. For the U.S. dataset, 80% of the target hosts, the estimation error is smaller using GeoBuD compared to CBG. The median of the location estimation error is of 144 km for GeoBuD, while of 228 km for CBG. In the W.E. dataset, it is 100 km and 137 km for GeoBuD and CBG respectively.

5.3 Upper Bounds on Distance Constraints

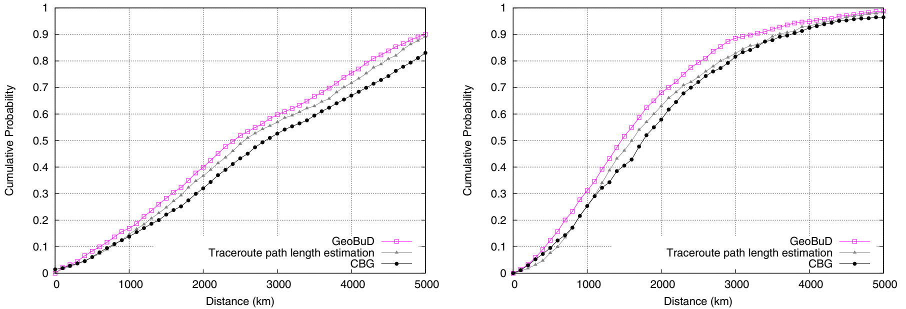
The fundamental idea of CBG relies on the controlled distance overestimation provided by the distance constraints. The goal is to overestimate in a controlled way so that distance constraints provide the smallest possible region that still encloses the target host. In practice, however, it is important to verify that distance constraints inferred between each landmark and the target hosts effectively provide an upper bound on the actual geographic distance between them. To evaluate whether CBG provides upper bounds on the actual distance, Figure 5 provides the cumulative distribution of the distances for each landmark-target pair.

Figure 5 compares the cumulative distribution of the estimated distances for each landmark-target pair using CBG, the estimation of the traceroute path length, and GeoBuD. The traceroute path length was computed by adding the geographic distances between the intermediate nodes along the traceroute which we were able to geolocate. For estimated distances larger than 1000 km, Figure 5 shows that CBG indeed provides



(a) U.S. dataset (b) Western Europe dataset

Fig. 4. Location estimation error for GeoBuD and CBG



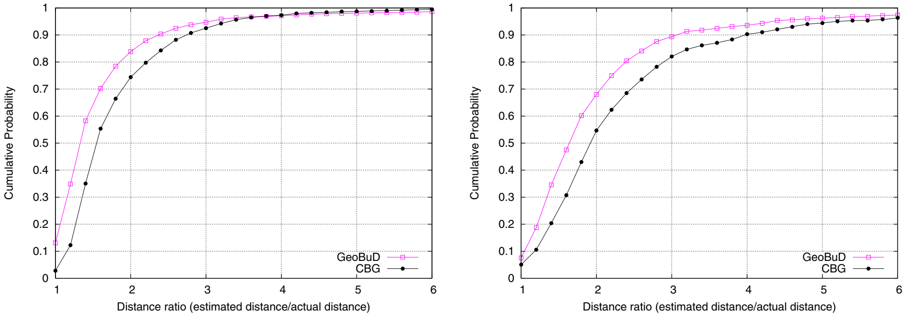
(a) U.S. dataset (b) Western Europe dataset

Fig. 5. Comparison of distance constraints and path length

an upper bound on the actual geographic distance, as even the estimated geographic length of the traceroute path is smaller. For distances smaller than 1000 km, CBG sometimes is close or below the estimated length of the traceroute path, as the estimated traceroute path length is also an upper bound on the actual geographic distance. Concerning GeoBuD, we observe in Figure 5 that it is a stricter upper bound on the distance than CBG or the estimated traceroute path length.

To understand why GeoBuD outperforms the original CBG, we need to recall how CBG transforms the delay into a distance constraint. For a given landmark, CBG overestimates the actual geographic distance by calibrating its transformation of the delay into a distance constraint by defining the value of b (see Equation (1)) based on the targets having the lowest delay measurement. This approach has the advantage of providing a conservative upper bound on the distance, as showed in Figure 5. However, its drawback compared to GeoBuD is of obtaining larger confidence regions.

In Figure 6 we plot the cumulative probability of the distance ratio over all landmark-target pairs, *i.e.* the ratio of the estimated distance (with CBG and GeoBuD) to the actual geographic distance. The purpose of Figure 6 is to study how each approach overestimates the actual distance. For instance, if we were to know that all estimated distances have a distance ratio larger than some value, then we could re-calibrate the estimated distance by dividing all distance estimates by this factor. Unfortunately, we can see on Figure 6 that there is a small fraction of the estimated distances that do not overestimate the actual distance (ratio = 1). In fact, 3% of the landmark-target pairs do not overestimate the actual distance for CBG, and 13% for GeoBuD (see Figure 6(a)). In Figure 6(b) we have 5% and 7% for CBG and GeoBuD respectively. If we were to perform this re-calibration for the hosts having a distance ratio of 1, these distance constraints would underestimate the actual distance, potentially leading to an empty confidence region. From Figure 6, we can also see that for a distance ratio smaller than 4, GeoBuD provides a tighter overestimation of the distance than CBG. This is another illustration of the improvement of GeoBuD compared to the CBG approach.



(a) U.S. dataset

(b) Western Europe dataset

Fig. 6. Cumulative probability of distance ratios

6 Conclusion

In this paper we have shown that estimating the buffering delays at intermediate hops along the traceroute between a landmark and a target host enables to improve the accuracy of the geolocation of Internet hosts. Based on traceroute measurements, we estimated the buffering delays at intermediate hops. By combining these buffering delay estimation with a multilateration technique (CBG [6]), we were able to shrink the confidence region where the target host is located. Results show that, with GeoBuD we obtain more accurate location estimation as well. As further work, we see the implementation of our approach as an on-line tool like GeoLIM [27]. We also aim at converging towards a confidence region as small as possible to provide better location estimation. We might also refine our estimation of the buffering delay by considering the potential existence of a bottleneck link on the path.

References

1. Qwerks, Inc., *WhereIsIP*, `http://www.jufsoft.com/whereisip/`.
2. MaxMind LLC, *GeoIP*, `http://www.maxmind.com/geoip/`.
3. Quova Inc., *GeoPoint*, `http://www.quova.com/`.
4. Venkata N. Padmanabhan and Lakshminarayanan Subramanian, "An investigation of geographic mapping techniques for Internet hosts," in *Proc. of the ACM SIGCOMM'2001*, San Diego, CA, USA, Aug. 2001.

5. Artur Ziviani, Serge Fdida, José Ferreira de Rezende, and Otto Carlos Muniz Bandeira Duarte, "Improving the accuracy of measurement-based geographic location of Internet hosts," *Computer Networks, Elsevier Science*, vol. 47, no. 4, pp. 503–523, Mar. 2005.
6. B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based geolocation of internet hosts," *IEEE/ACM Transactions on Networking*, 2006, to appear.
7. H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin, "The impact of routing policy on internet paths," in *Proc. of the IEEE INFOCOM'2001*, Anchorage, AK, USA, Apr. 2001.
8. Lakshminarayanan Subramanian, Venkata N. Padmanabhan, and Randy Katz, "Geographic properties of Internet routing," in *Proc. of USENIX 2002*, Monterey, CA, USA, June 2002.
9. H. Zheng, E. K. Lua, M. Pias, and T. Griffin, "Internet Routing Policies and Round-Trip-Times," in *Proc. of the Passive and Active Measurement Workshop – PAM'2005*, Boston, MA, USA, Apr. 2005.
10. Christopher Davis, Paul Vixie, Tim Goodwin, and Ian Dickinson, "A means for expressing location information in the domain name system," *Internet RFC 1876*, Jan. 1996.
11. University of Illinois at Urbana-Champaign, *IP Address to Latitude/Longitude*, `http://cello.cs.uiuc.edu/cgi-bin/slamm/ip2ll/`.
12. David Moore, Ram Periakaruppan, Jim Donohoe, and Kimberly Claffy, "Where in the world is netgeo.caida.org?," in *Proc. of the INET'2000*, Yokohama, Japan, July 2000.
13. M. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan, "Geographic locality of IP prefixes," in *Proc. of ACM/SIGCOMM Internet Measurement Conference – IMC 2005*, Berkeley, CA, USA, Oct. 2005.
14. *GeoURL*, `http://www.geourl.org/`.
15. *Net World Map*, `http://www.networldmap.com/`.
16. Visualware Inc., *VisualRoute*, `http://www.visualware.com/visualroute/`.
17. CAIDA, *GTrace*, `http://www.caida.org/tools/visualization/gtrace/`.
18. *Sarangworld Traceroute Project*, 2003, `http://www.sarangworld.com/TRACEROUTE/`.
19. Paramvir Bahl and Venkata N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. of the IEEE INFOCOM'2000*, Tel-Aviv, Israel, Mar. 2000.
20. C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and Piet van Mieghem, "Analysis of end-to-end delay measurements in Internet," in *Proc. of the Passive and Active Measurement Workshop – PAM'2002*, Fort Collins, CO, USA, Mar. 2002.
21. Artur Ziviani, Serge Fdida, José Ferreira de Rezende, and Otto Carlos Muniz Bandeira Duarte, "Toward a measurement-based geographic location service," in *Proc. of the Passive and Active Measurement Workshop – PAM'2004*, Antibes Juan-les-Pins, France, Apr. 2004, Lecture Notes in Computer Science (LNCS) 3015, pp. 43–52.
22. Stijn van Langen, Xiaoming Zhou, and Piet van Mieghem, "On the estimation of Internet distances using landmarks," in *Proc. of the International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking – NEW2AN'04*, St. Petersburg, Russia, Feb. 2004.
23. V. Jacobson, *Traceroure Software*, 1999, `ftp://ftp.ee.lbl.gov/traceroute.tar.Z`.
24. *PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services*, 2002, `http://www.planet-lab.org`.
25. *NLANR Active Measurement Project*, 1998, `http://watt.nlanr.net/`.
26. *RIPE Test Traffic Measurements*, 2000, `http://www.ripe.net/ttm/`.
27. *GeoLIM Project*, `http://planetlab-01.ipv6.lip6.fr:10000/cbg.php/`.
28. Neil Spring, Ratul Mahajan, and Thomas Anderson, "Quantifying the causes of path inflation," in *Proc. of the ACM SIGCOMM'2003*, Karlsruhe, Germany, Aug. 2003.

A Feedback Control Approach to Mitigating Mistreatment in Distributed Caching Groups*

Georgios Smaragdakis[1], Nikolaos Laoutaris[1,2], Ibrahim Matta[1],
Azer Bestavros[1], and Ioannis Stavrakakis[2]

[1] Computer Science Dept, Boston University, Boston, Massachusetts, USA
{gsmaragd, nlaout, matta, best}@cs.bu.edu
[2] Dept of Informatics and Telecommunications, University of Athens, Athens, Greece
istavrak@di.uoa.gr

Abstract. We consider distributed collaborative caching groups where individual members are autonomous and self-aware. Such groups have been emerging in many new overlay and peer-to-peer applications. In a recent work of ours, we considered distributed caching protocols where group members (nodes) cooperate to satisfy requests for information objects either locally or remotely from the group, or otherwise from the origin server. In such setting, we identified the problem of a node being *mistreated*, i.e., its access cost for fetching information objects becoming worse with cooperation than without. We identified two causes of mistreatment: (1) the use of a *common caching* scheme which controls whether a node should *not* rely on other nodes in the group by keeping its own local copy of the object once retrieved from the group; and (2) the *state interaction* that can take place when the miss-request streams from other nodes in the group are allowed to affect the state of the local replacement algorithm. We also showed that both these issues can be addressed by introducing two simple additional parameters that affect the caching behavior (the *reliance* and the *interaction* parameters). In this paper, we argue against a *static* rule-of-thumb policy of setting these parameters since the performance, in terms of average object access cost, depends on a multitude of system parameters (namely, group size, cache sizes, demand skewness, and distances). We then propose a *feedback control approach* to mitigating mistreatment in distributed caching groups. In our approach, a node independently emulates its performance as if it were acting selfishly and then adapts its reliance and interaction parameters in the direction of reducing its measured access cost below its emulated selfish cost. To ensure good convergence and stability properties, we use a (Proportional-Integral-Differential) PID-style controller. Our simulation results show that our controller adapts to the minimal access cost and outperforms static-parameter schemes.

Keywords: Cooperative Caching, Feedback Control, Simulation.

* A. Bestavros and I. Matta are supported in part by NSF grants EIA-0202067, ITR ANI-0205294, CNS-0524477 and CNS-0520166. I. Stavrakakis is supported in part by EU IST project CASCADAS. N. Laoutaris is supported by a Marie Curie Outgoing International Fellowship of the EU MOIF-CT-2005-007230.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 331–343, 2006.

1 Introduction

Background, Motivation, and Scope: Network applications often rely on distributed resources available within a cooperative grouping of nodes to ensure scalability and efficiency. Traditionally, such grouping of nodes is dictated by an overarching, common strategic goal. For example, nodes in a CDN such as Akamai or Speedera cooperate to optimize the performance of the overall network, whereas IGP routers in an Autonomous System (AS) cooperate to optimize routing within the AS.

More recently, however, new classes of network applications have emerged for which the grouping of nodes is more "ad hoc" in the sense that it is not dictated by organizational boundaries or strategic goals. Examples include the various overlay protocols [1, 2] and peer-to-peer (P2P) applications. Two distinctive features of such applications are (1) the fact that individual nodes are autonomous, and as such, their membership in a group is motivated solely by the selfish goal of *benefiting* from that group, and (2) group membership is warranted only as long as a node is interested in being part of the application or protocol, and as such, group membership is expected to be fluid. In light of these characteristics, an important question is this: *Are protocols and applications that rely on sharing of distributed resources appropriate for this new breed of ad-hoc node associations?*

As part of our recent work [3, 4], we studied this question for content networking applications, whereby the distributed resource being shared amongst a group of nodes is *storage*. In particular, we considered a group of nodes that store information objects and make them available to their local users as well as to remote nodes. A user's request is first received by the local node. If the requested object is stored locally, it is returned to the requesting user immediately, thereby incurring a minimal access cost. Otherwise, the requested object is searched for, and fetched from other nodes of the group, at a potentially higher access cost. If the object cannot be located anywhere in the group, it is retrieved from an origin server, which is assumed to be outside the group, thus incurring a maximal access cost. Contrary to most previous work in the field, we considered *selfish nodes*, *i.e.*, nodes that cater strictly and only to the minimization of the access cost for their local client population (disregarding any consequences for the performance of the group as a whole).

In [3, 4] we established the vulnerability of many *socially optimal* (SO) object replication/caching schemes to *mistreatment* problems. A mistreated node was defined as a node whose access cost under some cooperative scheme is higher than the corresponding minimal access cost that the node can guarantee for itself by being uncooperative. Unlike centrally designed/controlled groups where all constituent nodes have to abide by the ultimate goal of optimizing the social utility of the group, an autonomous, selfish node will not tolerate such a mistreatment. Indeed, the emergence of such mistreatments may cause selfish nodes to secede from the replication group, resulting in severe inefficiencies for both the individual users as well as the entire group.

Distributed Selfish Caching: Proactive replication strategies such as those studied in [3] are not practical in a highly dynamic content networking setting, which is likely to be the case for most of the Internet overlays and P2P applications we envision for a variety of reasons: (1) Fluid group membership makes it impractical for nodes to decide what to replicate based on what (and where) objects are replicated in the group. (2) Access patterns as well as access costs may be highly dynamic (due to bursty network/server load), necessitating that the selection of replicas and their placement be done continuously, which is not practical. (3) Both the identification of the appropriate re-invocation times [5] and the estimation of the non-stationary demands (or equivalently, the timescale for a stationarity assumption to hold) [6] are non-trivial problems. (4) Content objects may be dynamic and/or may expire, necessitating the use of "pull" (*i.e.*, on-demand caching) as opposed to "push" (*i.e.*, pro-active replication) approaches. Using on-demand caching is the most widely acceptable and natural solution to all of these issues because it requires no *a priori* knowledge of local/group demand patterns and, as a consequence, responds dynamically to changes in these patterns over time (*e.g.*, introduction of new objects, reduction in the popularity of older ones, *etc.*).

Therefore, in [4] we considered the problem of *Distributed Selfish Caching* (DSC), which could be seen as the *on-line* equivalent of the *Distributed Selfish Replication* (DSR) problem [3]. In DSC, we adopted an *object caching* model, whereby a node used demand-driven temporary storage of objects, combined with replacement. We examined the operational characteristics of a DSC group that can give rise to mistreatment problems and argued that simple parametric versions of already established protocols and mechanisms are capable of mitigating these problems. In this work we design a control theoretic framework for regulating the value of these parameters and thus adapting to fluid group conditions (varying group size, node capacities, delays and demand patterns). We thus significantly enhance our results from [4] which introduced these new control parameters but did not prescribe a complete method for regulating them in an adaptive manner.

Organization of the Paper: The rest of the paper is organized as follows. In Section 2 we describe our model of a distributed caching group. In Section 3 we demonstrate the causes of mistreatment in distributed caching groups. The design of a generic feedback controller for the mitigation of mistreatment is covered in Section 4. In this section, we also evaluate the performance of our adaptive scheme. Section 5 concludes the paper.

2 Model of a Distributed Caching Group

In this section we present the model of a distributed caching group that we consider in our study. Let o_i, $1 \leq i \leq N$, and v_j, $1 \leq j \leq n$, denote the ith unit-sized object and the jth node, and let $O = \{o_1, \ldots, o_N\}$ and $V = \{v_1, \ldots, v_n\}$ denote the corresponding sets. Node v_j is assumed to have storage capacity for up to C_j unit-sized objects, a total request rate λ_j (total number of requests per unit time, across all objects), and a demand described by a probability distribution over O, $\boldsymbol{p}_j = \{p_{1j}, \ldots, p_{Nj}\}$, where p_{ij} denotes the probability of object o_i being requested by the local users of node v_j. Successive requests are assumed to be independent and identically distributed.[1] For our numerical examples in later sections we will assume that the i^{th} most popular object is requested according to a generalized power-law distribution, i.e., with probability $p_i = K/i^{\alpha}$ (such distributions have been observed in many measured workloads [11, 13]).

Let t_l, t_r, t_s denote the access cost paid for fetching an object locally, remotely, or from the origin server, respectively, where $t_s > t_r > t_l$[2]; these costs can be interpreted either as delay costs for delivering an object to the requesting user or as bandwidth consumption costs for bringing the object from its initial location. User requests are serviced by the closest node that stores the requested object along the following chain: local node, group, origin server. Each node employs an object admission algorithm for storing (or not) objects retrieved remotely either from the group or from the origin server. Furthermore, each node employs a replacement algorithm for managing the content of its cache. In this work we focus on the Least Recently Used (LRU) replacement algorithm but we can obtain similar results under other replacement algorithms, such as Least Frequently Used (LFU) replacement algorithm (see also our previous work in [4]).

3 Mistreatment in Distributed Caching Groups

The examination of the operational characteristics of a group of nodes involved in a distributed caching solution enabled us to identify two key culprits for the emergence of mistreatment phenomena [4]: (1) the use of a *common caching scheme* across all the nodes of the group, irrespectively of the particular capabilities and characteristics of each individual one, and (2) the mutual *state interaction* between replacement algorithms running on different nodes.

3.1 Mistreatment Due to Common Scheme

The *common caching scheme* problem is a very generic vehicle for the manifestation of mistreatment. To understand it, one has first to observe that most of the work on cooperative caching has hinged on the fundamental assumption that all nodes in a cooperating group adopt a common caching scheme. We use the word "scheme" to refer to the combination of: (i) the employed *replacement algorithm*, (ii) the employed *request redirection algorithm*, and (iii) the employed

[1] The Independent Reference Model (IRM) [7] is commonly used to characterize cache access patterns [8, 9, 10, 11]. The impact of temporal correlations was shown in [6, 12] to be minuscule, especially under typical, Zipf-like object popularity profiles.

[2] The assumption that the access cost is the same across all node pairs in the group is made only for the sake of simplifying the presentation (those values can also be assumed as upper bounds of our analysis). Our results can be adapted easily to accommodate arbitrary inter-node distances.

object admission algorithm. Cases (i) and (ii) are more or less self-explanatory. Case (iii) refers to the decision of whether to cache locally an incoming object after a local miss. The problem here is that the adoption of a common scheme can be beneficial to some of the nodes of a group, but harmful to others, particularly to nodes that have special characteristics that make them "outliers". A simple case of an outlier, is a node that is situated further away from the center of the group, where most nodes lie. Here distance may have a topological/affine meaning (*e.g.*, number of hops, or propagation delay), or it may relate to dynamic performance characteristics (*e.g.*, variable throughput or latencies due to load conditions on network links or server nodes). Such an outlier node cannot rely on the other nodes for fetching objects at a small access cost, and thus prefers to keep local copies of all incoming objects. The rest of the nodes, however, as long as they are close enough to each other, prefer not to cache local copies of incoming objects that already exist elsewhere in the group. Since such objects can be fetched from remote nodes at a small access cost, it is better to preserve the local storage for keeping objects that do not exist in the group and, thus, must be fetched from the origin server at a high access cost.

Enforcing a common scheme under such a setting is bound to mistreat either the outlier node or the rest of the group. Consider the group depicted in *Figure* 1 in which $n-1$ nodes are clustered together, meaning that they are very close to each other ($t_r \to t_l \approx 0$), while there's also a single "outlier" node at distance t'_r from the cluster. The $n-1$ nodes would naturally employ a *Single Copy* (SC) scheme, *i.e.*, a scheme where there can be at most one copy of each distinct object in the group (e.g. LRU-SC [14]) in order to capitalize on their small remote access cost. From the previous discussion it should be clear that the best scheme for the outlier node would depend on t'_r. If $t'_r \to t_r$, the outlier should obviously follow LRU-SC and avoid duplicating objects that already exist elsewhere in the group. If $t'_r \gg t_r$, then the outlier should follow a *Multiple Copy* (MC) scheme, *i.e.*, a scheme where there can be multiple copies of the same object at different nodes — an example of an MC scheme is the LRU-MC. Under LRU-MC, if a node retrieves an object from a remote node in the group (or the origin server), then it stores a copy of it locally replacing an existing object if the cache is full, according to the LRU policy.

3.2 Mistreatment Due to State Interaction

The *state interaction* problem takes place through the so-called "remote hits". Consider nodes v, u and object o. A request for object o issued by a user of v that cannot be served at v but could be served at u is said to have incurred a *local miss* at v, but a *remote hit* at u. Consider now the implications of the remote hit at u. If u does not discriminate between hits due to local requests and hits due to remote requests, then the remote hit for object o will affect the state of the replacement algorithm in effect at u. If u is employing LRU replacement, then o will be brought to the top of the LRU list. If it employs LFU replacement, then its frequency will be increased, and so on with other replacement algorithms [15]. If the frequency of remote hits is sufficiently high, *e.g.*, because v has a much

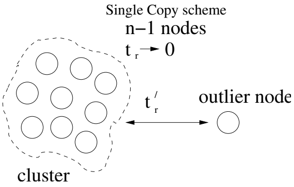


Fig. 1. An example of a group composed of a cluster of $n-1$ nodes and a unique outlier

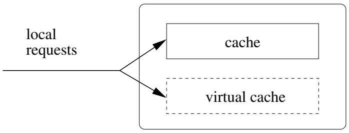


Fig. 2. Block diagram of a node equipped with a virtual cache

higher local request rate and thus sends an intense miss-stream to u, then there could be performance implications for the second: u's cache may get invaded by objects that follow v's demand, thereby depriving the users of u from valuable storage space for caching their own objects. This can lead to the mistreatment of u, whose cache is effectively "hijacked" by v.

4 Towards Mistreatment-Resilient Caching

From the exposition so far, it should be clear that there exist situations under which an inappropriate, or enforced, scheme may mistreat some of the nodes. While we have focused on detecting and analyzing two causes of mistreatment which appear to be important (namely, due to cache state interactions and the adoption of a common cache management scheme), it should be evident that mistreatments may well arise through other causes. For example, we have not investigated the possibility of mistreatment due to request re-routing [16], not to mention that there are vastly more parameter sets and combinations of schemes that cannot all be investigated exhaustively.

4.1 Design Disciplines

To address the above challenges, we first sketch a general framework for designing mistreatment-resilient schemes. We then apply this general framework to the two types of mistreatments that we have considered in this work. We target "open systems" in which group settings (*e.g.*, number of nodes, distances, demand patterns) change dynamically. In such systems it is not possible to address the mistreatment issue with predefined, fixed designs. Instead, we believe that *nodes should adjust their scheme dynamically so as to avoid or respond to mistreatment if and when it emerges.* To achieve this goal we argue that the following three requirements are necessary.

Detection Mechanism: This requirement is obvious but not trivially achievable when operating in a dynamic environment. *How can a node realize that it is being mistreated?* In our previous work on replication [3], a node compared its access cost under a given replication scheme with the guaranteed maximal access cost obtained through greedy local (GL) replication. This gave the node a "reference point" for a mistreatment test. In that game theoretic framework, we considered nodes that had *a priori* knowledge of their demand patterns, thus could easily compute their GL cost thresholds. In caching, however, demand patterns (even local ones) are not known *a priori*, nor are they stationary. Thus in our DSC setting, the nodes have to estimate and update their thresholds in an on-line manner. We believe that a promising approach for this is *emulation.* Figure 2 depicts a node equipped with an additional *virtual cache*, alongside its "real" cache that holds its objects. The virtual cache does not hold actual objects, but rather object identifiers. It is used for emulating the cache contents and the access cost under a scheme *different from* the one being currently employed by the node to manage its "real" cache under the same request sequence (notice that the input local request stream is copied to both caches). The basic idea is that *the virtual cache can be used for emulating the threshold cost that the node can guarantee for itself by employing a greedy scheme.*

Mitigation Mechanism: This requirement ensures that a node has a mechanism that allows it to react to mistreatment—a mechanism via which it is able to respond to the onset of mistreatment. In the context of the common scheme problem, the outlier should adjust its caching behavior according to its distance from the group. For this purpose, we introduce the LRU(q)-scheme, under which, objects that are fetched from the group are cached locally only with probability q; q will hereafter be referred to as the *reliance parameter*, capturing the amount of reliance that the node puts into being able to fetch objects efficiently from other nodes. In the context of the state interaction problem, one may define an *interaction parameter* p_s and the corresponding LRU(p_s) scheme, in which a remote hit is allowed to affect the local state with probability p_s, whereas it is denied such access with probability (1-p_s). As it will be demonstrated later on, nodes may avoid mistreatment by selecting appropriate values for these parameters according to the current operating conditions.

Control Scheme: In addition to the availability of a mistreatment mitigation mechanism (*e.g.,* LRU(q)), there needs to be a programmatic scheme for adapting the control variable(s) of that mechanism (*e.g.,* how to set the value of q). Since the optimal setting of these control variables depends heavily on a multitude of other time-varying parameters of the DSC system (*e.g.*, group size, storage capacities, demand patterns, distances), it is clear that there cannot be a simple (static) rule-of-thumb for optimally setting the control variables of the mitigation mechanism. To that end, dynamic feedback-based control becomes an attractive option.

To make the previous discussion more concrete, we now focus on the common scheme problem and demonstrate a mistreatment-resilient solution based on the previous three principle requirements. A similar solution can be developed for the state interaction problem.

4.2 Resilience to Common-Scheme-Induced Mistreatments

We start with a simple "hard-switch" solution that allows a node to change operating parameters by selecting between two alternative schemes. This can be achieved by using the virtual cache for emulating the LRU(q =1) scheme, capturing the case that the outlier node does not put any trust on the remote nodes for fetching objects and, thus, keeps copies of all incoming objects after local misses. Equipped with such a device, the outlier can calculate a running estimate of its threshold cost based on the objects it emulates as present in the virtual cache.[3] By comparing the access cost from sticking to the current scheme to the access cost obtained through the emulated scheme, the outlier can decide which one of the two schemes is more appropriate. For example, it may transit between the two extreme LRU(q) schemes–the LRU($q = 0$) scheme and the LRU($q = 1$) scheme. Figure 3 shows that the relative performance ranking of the two schemes depends on the distance from the group t'_r and that there is a value of t'_r for which the ranking changes.

A more efficient design can be obtained by manipulating the reliance parameter q at a finer scale. Indeed, there are situations in which intermediate values of q, $0 < q < 1$, are better than either $q = 0$ and $q = 1$ (see the LRU(0.1) and LRU(0.5) curves in Fig. 4). Consider two different values of the reliance parameter q_1 and q_2 such that $q_1 < q_2$. Figure 5 illustrates a typical behavior of the average object access cost under q_1 and q_2 as a function of the distance t'_r of the outlier node from its cooperative cluster. As discussed in the previous section, q_1 (q_2) will perform better with small (large) t'_r. In the remainder of this section, we present and evaluate a Proportional-Integral-Differential (PID) controller for controlling the value of q. This type of controller is known for its good convergence and stability properties (converges to a target value with zero error) [17, 18].

A node equipped with the PID controller maintains an Exponential Weighted Moving Average (EWMA) of the object access cost ($cost_{virtual}$) for the emulated greedy scheme. The virtual cache emulates an LRU($q = 1$)-scheme in which no remote fetches are considered, so as to avoid doubling the number of queries sent to remote nodes. Let $cost_q$ denote the EWMA of the object access cost of the employed LRU(q)-scheme in the actual cache of the node. Let *dist* denote the difference between the virtual access cost and the actual access cost, and let *diff* be the difference between two consecutive values of *dist*.

The PID controller adapts q proportionally to the magnitude of *diff*; a pseudocode for this process is provided in Algorithm 1. In [19], we argue that the access cost of a node equipped with this controller converges to a value which is lower than that of any scheme that employs a fixed q. We also provide an estimation of the converged value as a function controller parameters and other system characteristics.

Performance Evaluation: In order to evaluate our adaptive scheme, we compare its steady-state average access cost to the corresponding cost of one of the

[3] The outlier can include in the emulation the cost of remote fetches that would result from misses in the emulated cache contents; this would give it the exact access cost under the emulated scheme. A simpler approach would be to replace the access cost of remote fetches by that from the origin server and thus reduce the inter-node query traffic; this would give it an upper bound on the access cost under the emulated scheme.

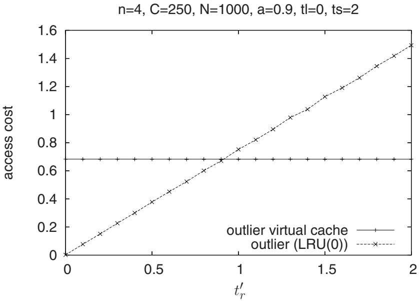


Fig. 3. Simulation results on the effect of the remote access cost t'_r on the access cost of the outlier node under the virtual cache and LRU(0) schemes

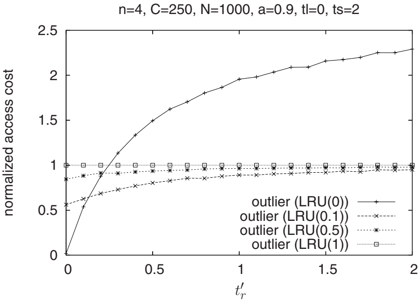


Fig. 4. Simulation results on the effect of the remote access cost t'_r on the normalized (by the virtual cost) access cost of the outlier node under different LRU(q) schemes

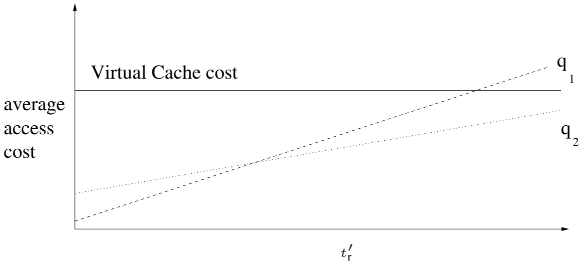


Fig. 5. Representative behavior of average object access cost as a function of the reliance parameter and distance of the outlier from the cluster

two extreme static schemes (LRU($q = 0$) or LRU($q = 1$)). Thus, we define the following performance metric:

$$minimum\ cost\ reduction\ (\%) \ = \ 100 \cdot \frac{cost_{static} \ - \ cost_{adaptive}}{cost_{static}} \tag{1}$$

where $cost_{adaptive}$ is the access cost of our adaptive mechanism, and $cost_{static}$ is the minimum cost of the two static schemes: $cost_{static} = min(\ cost(LRU(q = 0)), cost(LRU(q = 1))\ )$. This metric captures the minimum additional benefit that our adaptive scheme has over the previous static schemes. To capture the maximum additional benefit of our adaptive scheme (the optimistic case), we similarly define *maximum cost reduction* as in Eq. (1), where $cost_{static} = max(\ cost(LRU(q = 0)), cost(LRU(q = 1))\ )$.

We evaluate the performance of our PID-style feedback controller experimentally by considering a scenario in which the distance between the outlier node and the cooperative group (t'_r) changes according to the Modified Random

Algorithm 1 . Mitigation of mistreatment

```
$dist(t) \ = \ cost_{virtual}(t) - cost_q(t)$
$dist(t-1) \ = \ cost_{virtual}(t-1) - cost_q(t-1)$
$diff(t) \ = \ dist(t) \ - \ dist(t-1)$
$\sigma \ = \ sign(diff(t))$
if $q(t-1) \ \geq \ q(t-2)$ then
    $q(t) \ \leftarrow \ q(t-1) \ + \ \sigma \cdot \alpha_c \cdot |diff(t)| + \ \sigma \cdot \beta_c \cdot | \ |diff(t)| \ - \ |diff(t-1)| \ |$
else
    $q(t) \ \leftarrow \ q(t-1) \ - \ \sigma \cdot \alpha_c \cdot |diff(t)| - \ \sigma \cdot \beta_c \cdot | \ |diff(t)| \ - \ |diff(t-1)| \ |$
```

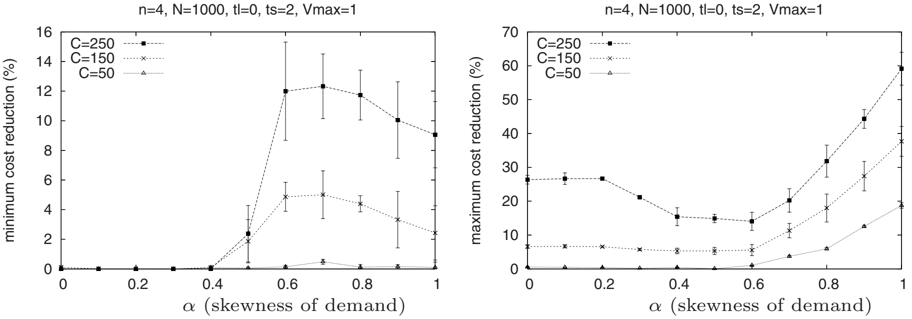


Fig. 6. Simulation results on the cost reduction that is achieved using our adaptive mechanism, (left): The minimum cost reduction, (right): The maximum cost reduction

Waypoint Model[4] [20]. The motivation for such a scenario comes from a wireless caching application [21]. A detailed description of the design of this experiment is provided in [19]. Figure 6 summarizes results we obtained under different cache sizes, demand skewness, and movement speed $V_{max} = 1$ distance units/time unit (similar results are observed under higher speeds as well). All experiments were repeated 10 times and we include 95^{th}-percentile confidence intervals in the graphs.

By employing our adaptive scheme, the outlier achieves a maximum cost reduction that can be up to 60% under skewed demand. The depicted profile of the maximum cost reduction curve can be explained as follows. The worst performance of the static schemes appears at the two extremes of skewness. Under uniform demand, $\alpha = 0$, we get the worst performance of the LRU(1) static scheme, whereas under highly skewed demand, $\alpha = 1$, we get the worst performance of the LRU(0) static scheme. In the intermediate region both static schemes provide for some level of compromise, and thus the ratio of the cost achieved by either scheme to the corresponding cost of the adaptive scheme becomes smaller than in the two extremes.

Turning our attention to the minimum cost reduction, we observe that it can be substantial under skewed demand, and disappears only under uniform demand

[4] This recent version fixes the non-stationarity of the original model, and thus provides better statistical confidence.

(such demand, however, is not typically observed in measured workloads [11]). The explanation of this behavior is as follows. At the two extreme cases of skewness, one of the static scheme reaches its best performance—under low skewed demand, the best static scheme is the LRU(0) and under high skewed demand the best static scheme is the LRU(1). Thus, the ratio of the cost achieved by the best static scheme and the corresponding cost of our adaptive scheme gets maximized in the intermediate region, in which neither of the static schemes can reach its best performance.

4.3 Resilience to State-Interaction-Induced Mistreatments

Immunizing a node against mistreatments that emerge from state interactions could be similarly achieved. The interaction parameter p_s can be controlled using schemes similar to those we considered above for the reliance parameter q. It is important to note that one may argue for *isolationism* (by permanently setting $p_s = 0$) as a simple approach to avoid state-interaction-induced mistreatments. This is not a viable solution. Specifically, by adopting an LRU($p_s = 0$) approach, a node is depriving itself from the opportunity of using miss streams from other nodes to improve the accuracy of LRU-based cache/no-cache decisions (assuming a uniform popularity profile for group members).

To conclude this section, we note that the approaches we presented above for mistreatment resilience may be viewed as “passive” or “end-to-end” in the sense that a node infers the onset of mistreatment *implicitly* by monitoring its utility function. As we alluded at the outset of this paper, for the emerging class of network applications for which grouping of nodes is “ad hoc” (*i.e.*, not dictated by organizational boundaries or strategic goals), this might be the only realistic solution. In particular, to understand “exactly how and exactly why” mistreatment is taking place would require the use of proactive measures (*e.g.*, monitoring/policing group member behaviors, measuring distances with pings, *etc.*), which would require group members to subscribe to some common services or to trust some common authority—both of which are not consistent with the autonomous nature (and the mutual distrust) of participating nodes.

5 Conclusions

We introduced a feedback control approach to mitigating mistreatment in distributed caching groups. Our approach controls the reliance and interaction parameters (q and p_s) by measuring the node’s current access cost under the current scheme and comparing it to the node’s emulated selfish access cost. By adapting q and p_s in the direction of moving the current access cost below the selfish cost, our PID-style controller reaps the benefits of cooperation whenever possible under the current system conditions (group size, cache sizes, demand skeweness, distances). Our simulation results confirm the premise of our adaptive (feedback) controller—it effectively adapts to the minimal access cost and even outperforms static controllers (where q and p_s are statically set to zero or one for no- or full-cooperation, respectively) under a wide range of system parameters.

To the best of our knowledge, this is the first attempt to use feedback control to ensure cooperation is always beneficial to users who are autonomous and selfish. Although we considered distributed caching, we believe similar feedback control can be successfully applied to other cooperative applications as well—we intend to investigate this in our future work.

References

1. Byers, J.W., Considine, J., Mitzenmacher, M., Rost, S.: Informed content delivery across adaptive overlay networks. IEEE/ACM Transactions on Networking **12**(5) (2004) 767–780
2. Cohen, E., Shenker, S.: Replication strategies in unstructured peer-to-peer networks. In: Proceedings of ACM SIGCOMM'02 Conference, Pittsburgh, PA, USA (2002)
3. Laoutaris, N., Telelis, O., Zissimopoulos, V., Stavrakakis, I.: Distributed selfish replication. IEEE Transactions on Parallel and Distributed Systems (2005) [accepted for publication].
4. Laoutaris, N., Smaragdakis, G., Bestavros, A., Stavrakakis, I.: Mistreatment in distributed caching groups: Causes and implications. In: Proceedings of IEEE Infocom, Barcelona, Spain (2006)
5. Loukopoulos, T., Lampsas, P., Ahmad, I.: Continuous replica placement schemes in distributed systems. In: Proceedings of the ACM ICS, Boston, MA (2005)
6. Jin, S., Bestavros, A.: Sources and Characteristics of Web Temporal Locality. In: Proceedings of IEEE/ACM Mascots'2000, San Fransisco, CA (2000)
7. Coffman, E.G., Denning, P.J.: Operating systems theory. Prentice-Hall (1973)
8. Arlitt, M.F., Williamson, C.L.: Web server workload characterization: the search for invariants. In: Proceedings of the 1996 ACM SIGMETRICS international conference on Measurement and modeling of computer systems. (1996) 126–137
9. Cao, P., Irani, S.: Cost-aware WWW proxy caching algorithms. In: Proceedings of USITS. (1997)
10. Young, N.: The k-server dual and loose competitiveness for paging. Algorithmica **11** (1994) 525–541
11. Breslau, L., Cao, P., Fan, L., Philips, G., Shenker, S.: Web caching and Zipf-like distributions: Evidence and implications. In: Proceedings of IEEE Infocom, New York (1999)
12. Psounis, K., Zhu, A., Prabhakar, B., Motwani, R.: Modeling correlations in web traces and implications for designing replacement policies. Computer Networks **45** (2004)
13. Mahanti, A., Williamson, C., Eager, D.: Traffic analysis of a web proxy caching hierarchy. IEEE Network **14**(3) (2000) 16–23
14. Fan, L., Cao, P., Almeida, J., Broder, A.Z.: Summary cache: a scalable wide-area web cache sharing protocol. IEEE/ACM Transactions on Networking **8**(3) (2000) 281–293
15. Podlipnig, S., Böszörmenyi, L.: A survey of web cache replacement strategies. ACM Computing Surveys **35**(4) (2003) 374–398
16. Pan, J., Hou, Y.T., Li, B.: An overview DNS-based server selection in content distribution networks. Computer Networks **43**(6) (2003)
17. Ogata, K.: Modern control engineering (4th ed.). Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2002)

18. Franklin, G.F., Powell, D.J., Emami-Naeini, A.: Feedback Control of Dynamic Systems (5th ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA (2005)
19. Laoutaris, N., Smaragdakis, G., Bestavros, A., Matta, I., Stavrakakis, I.: Distributed Selfish Caching. Technical Report BUCS-TR-2006-003, CS Department, Boston University (2006)
20. Lin, G., Noubir, G., Rajaraman, R.: Mobility models for ad hoc network simulation. In: Proceedings of IEEE Infocom, Hong Kong (2004)
21. Yin, L., Cao, G.: Supporting cooperative caching in ad hoc networks. In: Proceedings of IEEE Infocom, Hong Kong (2004)

Locality of Reference in an Hierarchy of Web Caches

Fernando Duarte, Fabrício Benevenuto,
Virgílio Almeida, and Jussara Almeida

Computer Science Department,
Federal University of Minas Gerais, Brazil
{fernando, fabricio, virgilio, jussara}@dcc.ufmg.br

Abstract. This work presents an extensive evaluation of the request filtering in hierarchy of proxy caches. Using the recently proposed ADF (Aggregation, Disaggregation and Filtering) model as well as entropy as metric for Web traffic characterization, we evaluate how locality of reference changes as the streams of requests pass through a hierarchy of caches. Moreover, we propose the use of average entropy for comparing the locality of reference of different streams and present how a proxy server can dynamically calculate the entropy of its incoming request stream.

Keywords: Web caching, locality of reference, entropy.

1 Introduction

The dramatic growth of network traffic and the increasing number of users are characteristics that have marked the phenomenon of the Web. The use of proxy servers has emerged as an efficient solution to increase the performance of Web systems, improving Web servers scalability and, reducing network traffic as well as the response time of user requests.

A proxy server can be seen as an intermediator of the traffic among clients and HTTP servers. When a proxy server sends a previously requested document to the clients, a copy of the document is stored in its local cache, so that future requests for this document can be directly obtained from the proxy server. These servers operate aggregating, disaggregating and filtering the request stream that passes through them. One can say they *aggregate* the arriving requests in an unique stream, which is processed using its local cache. Moreover, the proxy servers act as a *disaggregator* of traffic, distributing the arriving requests for different Web servers. When a stream of requests passes through a proxy, only the requests that could not be served from its cache are disaggregated towards the destination Web servers. In this context, one can say that a proxy acts as a request *filter*, allowing only the miss stream to be disaggregated to the rest of the Web.

Web caching is usually associated with a hierarchical organization. The browser cache, located in the user machine, is the lowest level of the hierarchy. The next

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 344–354, 2006.

level is composed of the caches of intranets, i.e., the proxy servers in universities and organizations. Going up in the hierarchy, there are regional proxies and so forth. A request that cannot be satisfied for a proxy is immediately sent to the proxy in the next level in the hierarchy, until it reaches the destination server.

The organization of an efficient cache hierarchy involves the study of the main properties of the streams of requests. Various questions related to *caching* require a deep study of how properties of the request streams change when they pass through the proxy servers. In this context, a vision of the Web traffic according to the effects of aggregation, filtering and disaggregation associated to the appropriated metrics bring a better understanding of the effects of locality of reference in an hierarchy of Web caches.

The study of locality of reference under this new perspective of the Web was initially considered in [5]. That work considered the use of entropy as metric to measure the locality of reference of request streams. In this work, we apply adaptations of this metric in a context of cache hierarchy, evaluating the impact that different cache replacement policies have on the locality of reference of the request streams. Moreover, we propose a methodology for calculating locality of reference dynamically. The main contributions of this paper are:

1. Performance evaluation of a cache hierarchy - This work provides an extensive performance evaluation of cache replacement policies in different levels of a hierarchy of caches. We measure traditional metrics such as hit ratio and compare these results with some recently proposed metrics for locality of reference. The experiments presented allow a better comprehension of how locality of reference changes as the streams of requests pass through an hierarchy of caches. This evaluation of the locality of reference can be used as a guide for designing of hierarchy of caches.
2. Metrics to locality of reference - We propose the average entropy to allow an HTTP server or a proxy server to perceive the variation of the locality of reference of request streams. Moreover, we present how entropy can be dynamically calculated by the Web components. We believe that capturing the notion of locality in real time can be helpful for constructing self-adaptive Web caching systems.

The rest of the paper is organized as follows. The next section presents related work. Section 3 introduces entropy and the new proposed metrics. Section 4 presents the experimental methodology used in this study. Our results are detailed in section 5. Conclusions and future work are offered in section 6.

2 Related Work

Although several different definitions are currently available [1, 6, 5], it is strongly accepted that the main aspects to locality of reference are *temporal correlations* in the request streams and the *popularity distribution* of requested objects [5, 4, 7]. This work focuses on the object popularity.

The study of locality of reference was motivated by the impact of this property on the performance of cache systems. These studies were the basis for the development of caching policies, inter-proxy communication protocols and prefetching algorithms [8].

The first attempt to characterize the impact of proxy caches on request streams is presented in [9]. Mahanti et al. studied how the temporal locality changes in different levels of a hierarchy of Web caches [6]. More recently, Williamson [10] evaluates the effectiveness of different caching policies in different levels of a hierarchy of Web caches. Whereas [10] only considered the filtering effects, [4, 5] introduced the study of two other transformations to which streams of references are submitted: aggregation and disaggregation. They grouped the three transformations into a model called ADF (Aggregation, Disaggregation and Filtering), and proposed and validated new metrics for analyzing temporal locality in a request stream moving through this model.

In this work, we evaluate the impact of locality of reference in the performance of a hierarchical caching system using the tools proposed in [5]. Moreover, we consider new forms of using the entropy proposed by [5] to analyze the locality of reference, providing a framework so that this metric can be dynamically calculated and applied to real environments.

3 Metrics for Locality of Reference

This section presents the metrics used in this paper to measure the locality of reference in streams of requests. In section 3.1 we present the concept of entropy. In section 3.2 we show an efficient form of calculating the entropy dynamically. In section 3.3, we propose the use of average entropy.

3.1 Entropy

The distribution of popularity of a set of requests usually is characterized by the *Zipf Law* [1, 3]. In general a Zipf-like distributions (the probability $P[i]$ of access the i-th most popular object is $P[i] = \frac{C}{i^\alpha}$, where α is a parameter and C a normalizing constant) has been used to approximate the popularity of objects in request streams in the Web. In this kind of distribution, the α coefficient is usually used as an indicator of the concentration of popularity of the request streams.

Recently, a more direct measure was proposed to evaluate the concentration of popularity of streams of requests, namely entropy [5]. The entropy $H(X)$ of a random variable X, taking n possible values with probability p_i, is calculated as follows:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

Note that $H(X)$ depends only on the probability of occurrence of the requests and the number n of different requests of the set. The maximum value ($H(X) = log_2 n$) is reached when the requests have the same probability ($p_i = 1/n, \forall i$), and the minimum value ($H(X) = 0$) occurs when only one object concentrates all references ($p_i = 1$, $p_j = 0$ for $i \neq j$). Thus, for sets with the same number of requests, the higher the value of the entropy, the smaller the concentration of popularity in few objects.

3.2 Calculating Entropy Dynamically

We now propose a technique to dynamically compute the entropy present in a request stream. Calculating entropy dynamically allows an online analysis of locality of reference, which can be used for making operational decisions. For instance, a proxy server can vary some parameters of its configuration based on the variations that occur in the locality of reference of the arriving request stream.

In order to use the definition of entropy, proposed and validated in [5], in real environments, its value must be measured in a incremental way, being recalculated at each new arriving request. Previous works [5], have computed entropy for a set of requests, in which the number of requests was known *a priori*.

Expanding the equation 1, we find a practical and dynamic way to calculate the entropy. Let n_t be the total number of requests that have already arrived at the proxy and n_i be the number of references for the object i in that set, then p_i can be estimated as n_i/n_t. The entropy can be calculated as:

$$H(X) = \log_2 n_t - \frac{1}{n_t} \sum_{i=1}^{n} n_i log_2 n_i \tag{2}$$

Using equation 2, the entropy can be dynamically calculated keeping up to date the value of n_t and the value of the sum $S = \sum_{i=1}^{n} n_i \log_2 n_i$ for each new arriving request.

3.3 Average Entropy

The normalized entropy was proposed to compare the entropy of sets with different number of requests [5]. This normalization is based on the highest possible value for the entropy of the set of requests. Considering n as the number of distinct requests, the normalized entropy $H^n(X)$ is defined as:

$$H^n(X) = \frac{H(X)}{log_2 n} \tag{3}$$

Nevertheless, when we deal with sets of requests of equal sizes, the entropies of these sets can be compared directly, being unnecessary the normalization presented in the equation 3. Based on this observation, we propose the average entropy. We calculate the entropy of a set of requests by considering a window of m requests at a time. The window moves one request at a time, and a new entropy value is calculated. At the end, after covering the whole set of requests, we average the entropy values computed for all request windows.

Considering a window of size m, a sequence with a total of n_t requests and $H(X_{[i,j]})$ as the value of the entropy of the window that contains the interval of the i-th until the j-th request, we define the average entropy $H^m(X)$ as:

$$H^m(X) = \frac{\sum_{i=0}^{n_t-m} H(X_{[i+1,m+i]})}{n_t - m + 1} \tag{4}$$

In order to use the notion of locality of reference in a real environment such as in a Web server or in a proxy server, one needs to evaluate the locality of a certain sample of the requests that arrive at the server. In this context the average entropy, associated to the dynamically calculation of the entropy, emerge as adjusted metrics to capture the variation of popularity of the stream of requests that arrives at the servers. This can be done, for instance, by comparing the entropy of the last window with the value of the average entropy.

The size of the window for the calculation of the entropy can impact the analysis of the popularity concentration. For instance, if the window is small, the obtained entropy captures the locality of a small sample, which cannot represent correctly the popularity of the flow. On the other hand, if the size of the window is relatively high, the variation of popularity is less noticeable. We suggest that, to compare different streams of requests it is necessary that the size of the window to be of the same order of magnitude of the total of requests of the streams.

4 Experimental Methodology

This section describes the methodology used in our study. A simulator of a hierarchy of caches was built, organized as showed in figure 1 (left). This figure presents a two-level caching system, with two caches (children) on the first level and one cache (parent) on the second one. The requests made by the users are received directly by the caches in the first level, whereas the requests that cannot be satisfied in this level are aggregated forming a request stream, which is forwarded to the second level cache. There is no interaction between the first level caches.

In order to better understand the effects of the locality of reference in the hierarchy of caches, we use the ADF model [5]. This model represents the Web through a graph where the vertices are points where the request streams can be modified, and the edges are connections among these points. The vertices in the graph are of three different kinds, depending of which effect they cause in the Web traffic: Aggregation (A), Disaggregation (D) and Filtering (F).

Figure 1 (right) shows the representation of the cache hierarchy used in our experiments using the ADF model. The caches of the first level function as points of aggregation of the user requests. These caches also apply a filtering transformation on the streams of requests. The streams of missed requests that are forwarded by the first level caches are aggregated and again, are filtered in second level cache, where they are finally disaggregated to the Web servers.

In the simulations, we evaluate the behavior of the average entropy when streams of requests pass through the hierarchy of caches, varying the size of these caches from $1MB$ to $16GB$. To choose the size of the window used in the calculation of the average entropy, several experiments were executed varying the window size. The difference of the results obtained for window sizes with

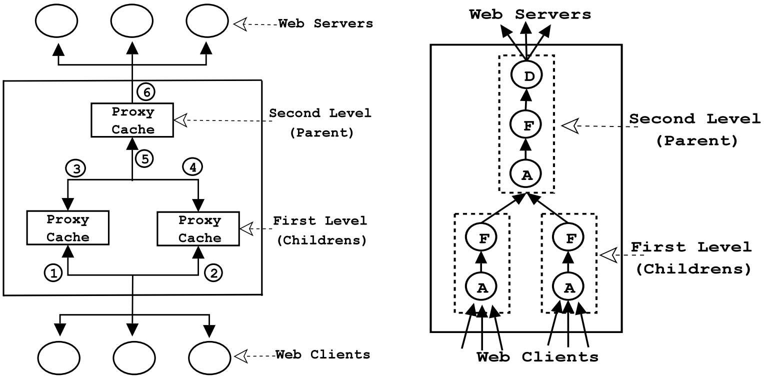


Fig. 1. System of hierarchy caches: ISP overview(*left*), ADF model(*right*)

order of magnitude 100,000 varies very little. Accordingly to it, this value was chosen for the experiments.

Four cache replacement policies were considered: LRU, LFU-Aging, GD-Size and LRU-Threshold. We evaluate the impact of these policies combined in the different levels of the hierarchy. For a comprehensive description of several replacement cache policies, see reference [8].

4.1 Workload Characteristics

This section presents the main characteristics of the logs used in the experiments. These logs were obtained from a Brazilian ISP[1], with a hierarchical caching system similar to the one presented in figure 1. We obtained logs of two machines of the first level of this hierarchy, which we call *Pop-1* and *Pop-2*. The main workload characteristics are presented in Table 1. The logs of the days Oct 16-17, 2001 were used to warm the caches whereas the measurements of entropy and hit ratio were obtained with logs of the days Oct 18-19, 2001.

Note that the number of different objects represents about 26% of the total number of requests in all four logs. From this percentage, about 69% are documents with only one reference (*1-timers*). Moreover, our workloads contain mostly small objects. As show in Table 1, the 3° quartile of the distribution of file sizes is under $3KB$. Nevertheless, some objects are relatively large for Web documents, which explains the coefficient of variation of the distributions of file sizes being relatively high.

5 Experimental Results

This section presents the results of the simulation of the hierarchy of caches illustrated in figure 1. The average entropy was measured in the points numbered in the figure, which are the points where we perceive the effect of filtering,

[1] POP-MG provides Internet access to incorporated customers and university users.

aggregation and disaggregation. For each caching polices LRU, LFU-Aging and GD-Size used in the caches at first level of the hierarchy; we evaluated different caching policies in the second level cache. The hit ratio and average entropy are calculated for each cache. Individual caches are identified as *child R*, *child L* and *Parent* for the first and second levels, respectively. Figures 2, 3 and 4 show the hit ratio and average entropy as the cache size increases.

Comparing the effectiveness of the first-level cache with the second-level cache, one can see that the first-level caches always get higher hit ratio. Comparing the entropy of the streams of requests before the hierarchy of caches with the entropy after the first level of caches, we verify that this occurs because the filtering of the first level caches absorbs part of the locality of reference and generates a stream of requests with smaller concentration of popularity for the second-level cache. The larger the first level caches, the higher is the filtering effect perceived. Thus, in some cases, the hit ratio of the second level cache decreases with the increase of the cache size at the first level. Moreover, as discussed in [2], the cache hit ratio becomes stabilized and reaches its maximum value when the cache is able to store all distinct objects. In our experiments, the maximum hit ratio for the first and second levels of the hierarchy occurs when the cache size is approximately 4GB.

We next discuss the variations of locality of reference comparing the entropy as the stream of requests pass through the hierarchy. We verify that the filtering diminishes the popularity when we compare the entropy of points 1 and 2 with the entropy of points 3 and 4, and the entropy of point 5 with the entropy of point 6. Moreover, we notice that the aggregation in point 5 diminishes the entropy, increasing the concentration of popularity. The entropy in points 3, 4, 5 and 6 grows until stabilizing as the cache sizes increases. This occurs when the caches are able to store all distinct objects, and the entropy tends to its upper bound, which indicates a sequence without popularity.

Table 1. Workload Characteristics

Item	**Pop-1**	**Pop-2**	**Pop-1**	**Pop-2**
Start Date	10/16/01	10/16/01	10/18/01	10/18/01
Duration (# days)	2	2	2	2
# requests	882,639	908,317	902,998	919,541
Distinct objects	234,663	246,560	238,880	237,290
1-timers	161,646	173,796	164,011	164,878
Workload Size (MB)	3,865	4,220	3,974	4,213
Smallest object	0	0	0	0
Largest object (MB)	33.13	41.75	29.61	49.70
Average Size (KB)	4.48	4.76	4.51	4.69
1° **Quartile (Bytes)**	365	372	364	371
Median (Bytes)	757	746	1,392	778
3° **Quartile (Bytes)**	2,690	2,698	2,576	2,571
Coefficient of Variation	16.52	29.29	14.22	19.62
Average Entropy	14.64	14.32	13.78	13.92

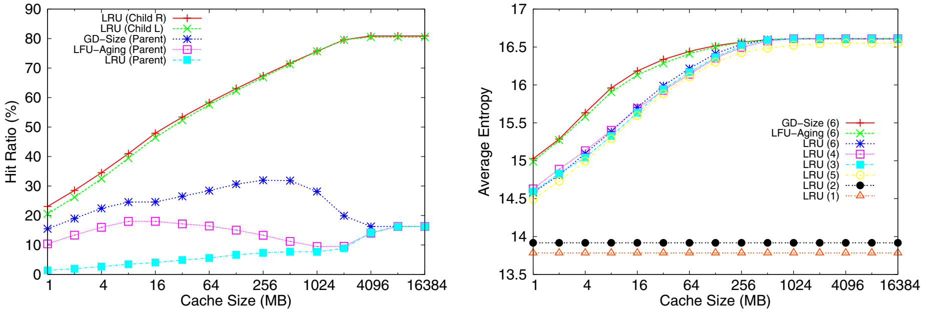


Fig. 2. LRU on the First Level - Hit Ratio and Average Entropy

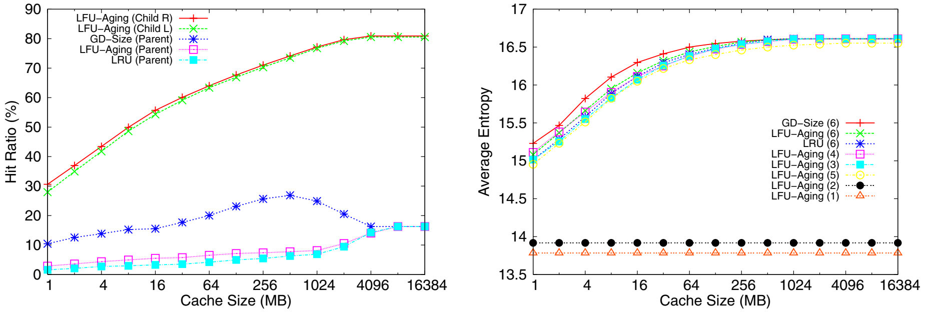


Fig. 3. LFU-Aging on the First Level - Hit Ratio and Average Entropy

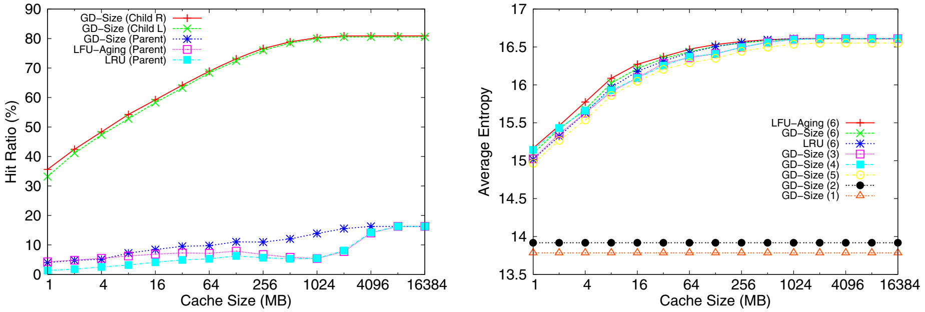


Fig. 4. GD-Size on the First Level - Hit Ratio and Average Entropy

Figure 5 shows the hit ratio and the average entropy when LRU-Threshold is used as the caching policy in the first level caches, storing just fewer files, with sizes smaller than $4KB$. Table 1 shows that this value is greater that the 3° quartile of the file size distribution for all logs, which indicates that most of objects can be stored in the first level, leaving only the largest objects to the second-level cache. Note that the sizes of the first level caches are large enough

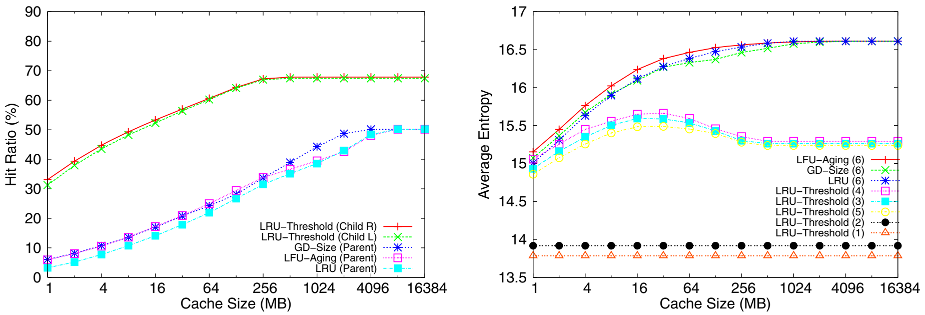


Fig. 5. LRU-Threshold on the First Level - Hit Ratio and Average Entropy

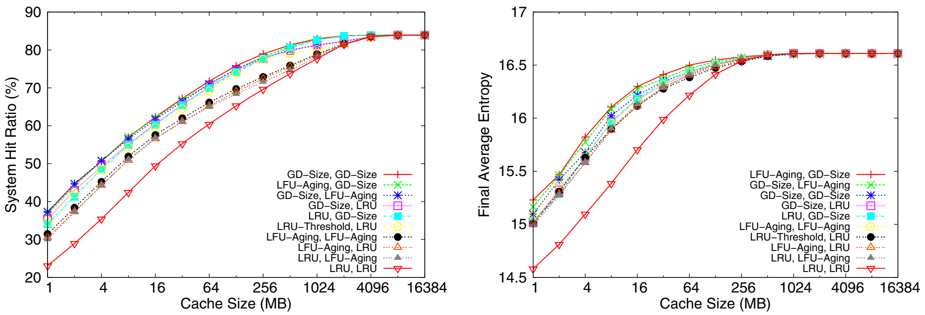


Fig. 6. Comparison among different configurations of the hierarchy of caches - Hit Ratio and Final Average Entropy

to hold all the objects smaller than $4KB$, the request stream that leaves these caches contains only references to objects smaller than $4KB$ that are 1-timers and to larger objects. Thus, the larger objects become relatively popular at the second-level cache, decreasing the entropy.

The relationship between entropy and hit ratio can be analyzed by observing the graphs in Figure 5. The reduction of the entropy in point 5, between the cache sizes 64 MB and 256 MB, has direct implication in the hit ratio of second-level cache. Until this point the LFU-Aging gets better hit ratio and, from this point on, GD-Size obtained the best result. This effect suggests that variations in the entropy can be used to dynamically configure caching policies in an hierarchy of caches. This is subject for future work.

In order to evaluate the performance of the hierarchy of caches as a whole, we simulated different configurations of caching policies. We consider the best configuration as the one that filters the concentration of popularity the most, i.e., the one which has the largest entropy in the stream of requests leaving the hierarchy. Figure 6 shows the final entropy and the hit ratio for some configurations of caching policies. The combination of LFU-Aging in the first-level caches and GD-Size in the second-level cache produced the best results, whereas the use of LRU in all caches performs the worst. Note that although the configuration with GD-Size in the two levels produced the best hit ratio, this configuration did not provide the best final entropy. This happens because these policies keep the smaller objects in the cache, thus increasing the hit ratio, but discarding bigger objects with some popularity. Therefore, this kind of policy does not act directly on the locality of reference.

6 Conclusions and Future Work

We use the ADF (Aggregation, Disaggregation and Filtering) model and entropy to evaluate the effects that the locality of reference of a request streams suffer as they pass through a hierarchy of caches. The proposed metrics are able to capture online the locality of reference at any component of the Web hierarchy. We believe that the notion of locality can be used for operational decisions and thus, it can be useful to construct automated Web caching systems. Our results show how the transformations of aggregation, filtering and disaggregation act in the locality of reference and the impact of these operations into the performance of a hierarchy of caches. In general, filtering on the first-level caches is more effective than filtering on the second-level cache, since the request streams leaving the first-level caches have lower entropy. However, the aggregation of the outgoing first-level request streams decrease the entropy of the stream offered to the second-level cache, which provides an opportunity for a better hit ratio on that cache. Furthermore, our results show that heterogeneous configurations of caching policies take advantage of the reference locality.

Directions for future work include to explore dynamic and average entropy in proxy servers and to develop a model for hierarchical caching system in which the caching policies for the different levels of this hierarchy can dynamically be modified, based on variations of the entropy of the requests that arrive at the caching system.

References

1. V. Almeida, A. Bestavros, M. Crovella, and A. Oliveira. Characterizing Reference Locality in the WWW. In *Proc. of PDIS*, December 1996.
2. F. Benevenuto, F. Duarte, V. Almeida, and J. Almeida. Web Cache Replacement Policies: Properties, Limitations and Implications. In *Proc. of Latin American Web Congress*, November 2005.
3. L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proc. of IEEE Infocom*, April 1999.
4. R. Fonseca, V. Almeida, and M. Crovella. Locality in a Web of Streams. *Communications of the ACM*, 48(1):82–88, January 2005.
5. R. Fonseca, V. Almeida, M. Crovella, and B. Abrahao. On the Intrinsic Locality Properties of Web Reference Streams. In *Proc. of IEEE Infocom*, May 2003.
6. A. Mahanti, D. Eager, and C. Williamson. Temporal Locality and its Impact on Web Proxy Cache Performance. *Performance Evaluation Journal: Special Issue on Internet Performance Modelling*, 42(2/3):187–203, September 2000.

7. S. Vanichpun and A. Makowski. Comparing Strength of Locality of Reference - Popularity, Majorization, and Some Folk Theorems. In *Proc. of IEEE Infocom*, March 2004.
8. J. Wang. A Survey of Web Caching Schemes for the Internet. *ACM Computer Communication Review*, 25(9):36–46, 1999.
9. D. Weikle, S. Mckee, and W. Wulf. Cache as Filters: A New Approach to Cache Analysis. In *Proc. of MASCOTS*, July 1998.
10. C. Williamson. On Filter Effects in Web Caching Hierarchies. *ACM Transactions on Internet Technology*, 2(1):47–77, February 2002.

DMTP: Controlling Spam Through Message Delivery Differentiation

Zhenhai Duan[1], Yingfei Dong[2], and Kartik Gopalan[1]

[1] Dept. of Computer Science, Florida State University, Tallahassee, FL 32309
Tel.: 01-850-645-1561; Fax: 01-850-644-0058
{duan, kartik}@cs.fsu.edu
[2] Dept. of Electrical Engineering, Univ. of Hawaii, 2540 Dole St., Honolulu, HI 96822
yingfei@hawaii.edu

Abstract. Unsolicited commercial email, commonly known as spam, has become a pressing problem in today's Internet. In this paper we re-examine the architectural foundations of the current email delivery system that are responsible for the proliferation of email spam. We argue that the difficulties in controlling spam stem from the fact that the current email system is fundamentally sender-driven and distinctly lacks receiver control over email delivery. Based on these observations we propose a Differentiated Mail Transfer Protocol (DMTP), which grants receivers greater control over how messages from different classes of senders should be delivered on the Internet. In addition, we also develop a formal mathematical model to study the effectiveness of DMTP in controlling spam. Through numerical experiments we demonstrate that DMTP can effectively reduce the maximum revenue that a spammer can gather; moreover, compared to the current SMTP-based email system, the proposed email system can force spammers to stay online for longer periods of time, which may significantly improve the performance of various real-time blacklists of spammers. Furthermore, DMTP provides an incremental deployment path from the current SMTP-based system in today's Internet.

Keywords: Email Spam, Unwanted Internet Traffic, SMTP, DMTP.

1 Introduction

Unsolicited commercial email, commonly known as *spam*, is a pressing problem on the Internet. In addition to undermining the usability of the current email system, spam also costs industry billions of dollars each year [7, 19]. In response, the networking research and industrial communities have proposed a large number of anti-spam countermeasures, including numerous email spam filters [9, 17], sender authentication schemes [3, 14], and sender-discouragement mechanisms (to increase the cost of sending email such as paid email) [8, 12]. Some of the schemes have even been extensively deployed on the Internet. On the other hand, despite these anti-spam efforts, in recent times the proportion of email spam seen on the Internet has been continuously on the rise.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 355–367, 2006.

1.1 Why Is It so Hard to Control Spam?

The current email system uses the Simple Mail Transfer Protocol (SMTP) to deliver messages from a sender to a receiver [13]. While simple, such a system also provides an ideal platform for spammers to act as parasites. It is our contention that, in order to effectively control spam, we must design and deploy an email delivery system that can proactively resist spam in the first place. As a first step toward this goal, in this paper we examine the architectural aspects of the current email system that are responsible for the proliferation of spam. We propose a Differentiated Mail Transfer Protocol (DMTP) that attempts to overcome these limitations based on the following three key insights.

Moving to a receiver-driven model: First, the current email system is fundamentally sender-driven and distinctly lacks receiver control over the message delivery mechanism. For example, in the current SMTP-based email system, any user can send an email to another at will, regardless of whether or not the receiver is willing to accept the message. In the early days of the Internet development, this was not a big problem as people on the network largely trusted each other. However, since the commercialization of the Internet in the mid-1990s, the nature of the Internet community has changed. It has become less trustworthy, and the emergence of email spam is one of the most notable examples of this change. In order to effectively address the issue of spam in the untrustworthy Internet, we argue that *the email architecture should provide receivers with greater control over if and when a message should be delivered to them.*

Eliminating economy of scale: Secondly, volume is the most crucial factor in making email spam a profitable business. In order to squeeze spammers out of business, we must eradicate the economy of scale they rely on. However, in the current email system, the sending rate of spam is, to a large extent, only constrained by the processing power and network connectivity of spammers' own mail servers, of which the spammers have complete control. Nowadays, with increasingly-powerful (and cheaper) PCs and ubiquitous high-speed Internet access, spammers can push out a deluge of spam within a very short period of time, making spamming profitable because of the economy of scale. We contend that *the email architecture should intrinsically regulate the sending rate of emails from individual senders, ideally under the control of email receivers*, in order to restrain spam.

Increasing accountability: Lastly, the current email system makes it hard to hold spammers accountable for spamming. Spammers can vanish (go offline) immediately after pushing a deluge of spam to receivers, which can be done within a very short period of time. This makes it quick and easy for spammers to hide their identities and provides spammers with the flexibility to frequently change their locations and/or Internet service providers—complicating the effort to filter spam based on the IP addresses of sender mail servers, such as using various real-time blacklists (RBLs) [17]. We argue that in order to hold spammers accountable and to make RBLs more effective, *the email architecture must force spammers to stay online for longer periods of time.*

1.2 Contributions of This Paper

Based on these observations we propose a Differentiated Mail Transfer Protocol (DMTP) as a countermeasure to the spam problem. A key feature of DMTP is that it grants receivers greater control over the message delivery mechanism. In DMTP, a receiver can classify senders into different classes and treat the delivery of messages from each class differently. For example, although regular contacts of a receiver can directly send messages to the receiver, unknown senders need to store messages in the *senders' own mail servers*. Such messages are only retrieved by the receiver *if and when* she wishes to do so.

DMTP provides us with several important advantages in controlling spam: 1) the delivery rate of spam is determined by the spam retrieval behavior of receivers instead of being controlled by spammers; 2) spammers are forced to stay online for longer periods of time (because the sending rate of spam is regulated by the spam retrieval rate of receivers), which can significantly improve the performance of RBLs; 3) regular correspondents of a receiver do not need to make any extra effort to communicate with the receiver—correspondence from regular contacts is handled in the same manner as in the current SMTP-based email system; 4) DMTP can be easily deployed on the Internet incrementally.

In this paper we present the design of DMTP and formally model its effectiveness in controlling spam. Through numerical analyses we show that DMTP can significantly reduce the maximum revenue that a spammer can obtain. In addition, a spammer has to stay online for a much longer period of time in order to obtain the maximum revenue.

The remainder of the paper is organized as follows. In Section 2 we re-examine two common traffic delivery models on the Internet: sender push vs. receiver pull, and discuss their implications on controlling spam. In Section 3 we present the design of DMTP, which employs a variant of the receiver-pull model. We formally model the effectiveness of DMTP in controlling spam and perform numerical analyses in Section 4. In Section 5 we discuss practical deployment issues of DMTP on the Internet. After describing related work in Section 6, we conclude the paper and outline our ongoing work in Section 7.

2 Push vs. Pull: Implications of Protocol Design Choice

Asynchronous messages like email are delivered on the Internet primarily using two different models: *sender-push and receiver-pull* (or a combination of the two) [5]. In this section we discuss the implications of the two models on controlling unwanted traffic on the Internet and illustrate that the receiver-pull model has several important advantages in discouraging unwanted Internet traffic such as email spam. In light of these advantages, in the next section we develop a new email delivery protocol based on the receiver-pull model.

The two models differ in who initiates the message delivery process. In the sender-push model, senders control the delivery of traffic, and receivers passively accept whatever the senders push to them. The current SMTP-based email delivery system is a typical example of this model. In contrast, the receiver-pull model grants receivers the control over if and when they want to retrieve data from the senders. In this model, senders can only prepare the data but they cannot push the data to receivers. Examples of the receiver-pull model include the HTTP-based web access services and the FTP-based file transfers.

While both simple and convenient, the sender-push model has a big disadvantage in controlling unwanted Internet traffic: in this model it is senders who completely control *what* messages are delivered and *when* the messages are delivered. Receivers do not have the knowledge of either what messages they will receive or when the messages will be received. Receivers have to receive the entire messages before processing or discarding the messages. Moreover, senders can vanish immediately after the messages are pushed out.

By contrast, the receiver-pull model comes with several appealing advantages because it grants receivers greater control over the message delivery mechanism. It takes advantage of the fact that receivers have more reliable knowledge of what traffic they want to receive. In this model, receivers have the freedom to first determine the reputation of the senders (and their own level of interest in the contents) *before* they actually request the content. Moreover, it becomes the responsibility of senders to store and manage the messages till the receivers are ready to retrieve the messages. This forces malicious senders to stay online and reveal their identities for larger windows of time.

A reasonable concern with the receiver-pull model is that it may increase the cost of sending messages for malicious as well as legitimate senders. We show in the next section that, using simple design optimizations, we can easily lower the sending cost for legitimate senders while still retaining the benefits of the receiver-pull model. In summary, although the receiver-pull model may result in slightly greater protocol complexity, it can greatly help to simplify the control of unwanted traffic such as spam on the Internet, and should be considered early in the design phase of any communication system.

3 DMTP: A Differentiated Mail Transfer Protocol

DMTP is designed based on a variant of the receiver-pull model, where senders are allowed to first express an intent to send message to a receiver via a small intention message. If the receiver happens to be interested, she contacts the sender and retrieves the content message. Figure 1 illustrates the architecture of the new email delivery system. The new system extends the current SMTP

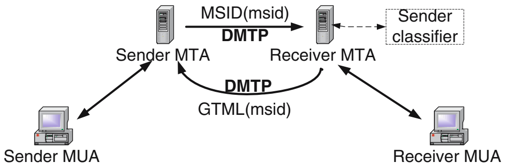


Fig. 1. Illustration of DTMP-based email system

Table 1. News commands/reply code defined in DMTP

Commands/Replies	Explanation
MSID	For SMTA to inform RMTA the *msid* of a message
GTML	For RMTA to retrieve a message from SMTA
253	For RMTA to inform SMTA to send *msid* (MSID) instead of messages (DATA)

protocol [13] by adding just two new commands–MSID and GTML, and one new reply code–253 (see Table 1). All the commands and reply codes in SMTP are also supported in the new system. We will explain the new commands and reply code in the following.

3.1 Differentiating Message Delivery

As discussed in the last section the receiver-pull model increases the cost of sending messages for both malicious and legitimate senders. To address this issue DMTP is designed to support a hybrid email delivery system where both the sender-push and receiver-pull models can be employed. Specifically, each receiver can classify email senders into three disjoint classes and treat the delivery of messages from each of them differently: 1) *well-known spammers*, whose messages will be directly rejected; 2) *regular contacts*, whose messages can be directly pushed from the senders to the receiver using the current SMTP protocol; and 3) *unclassified senders*–senders that are neither well-known spammers nor regular contacts. Unlike regular contacts, unclassified senders cannot directly push a message in its entirety to the receiver. Such messages need to be stored and managed by the *senders' mail servers*, and only the envelopes of the messages can be directly delivered to the receiver as notifications of pending messages.

Senders can be defined at the granularity of email addresses as well as IP addresses (and domain names) of sender mail servers. Given that it is easy to fake email addresses in the current Internet, we envision that sender classification will be performed at the granularity of IP addresses when DMTP is first deployed.

Fig. 2 summarizes the algorithm of handling message delivery requests at a DMTP receiver. In the figure we have assumed that the sender classification is only supported at the IP addresses (and domain names) level. Sender classification defined at the email address level can be easily incorporated into the algorithm. In the rest of this section we focus on the handling of messages from unclassified senders. The handling of messages from well-know spammers and regular contacts is the same as in the current practice [13, 17], and we omit the description.

3.2 Unclassified Sender: Message Composition and Receiver Notification

Like in the current email system, an (unclassified) sender uses a Mail User Agent (MUA) to compose outgoing messages [13]. After a message is composed by the sender, the sender delivers the message to the sender Mail Transfer Agent

```
Require: SPC: well-known spammer class;
Require: RCC: regular contact class;
Receiving TCP session open request on port 25;
ip = Get IP address of sender mail server;
if (ip ∈ SPC) then
    /* well-known spammers */
    reply with 550 (to decline TCP session opening request);
    close TCP session;
else if (ip ∈ RCC) then
    /* regular contacts */
    reply with 220 (to accept TCP session opening request);
    proceed as if SMTP used;
else
    /* unclassified senders */
    reply with 253 (see Table 1);
    accept MSID command;
    reject DATA command;
end if
```

Fig. 2. Algorithm for receivers to handle message delivery requests in DMTP

(MTA). For simplicity, we refer to a sender MTA server as an SMTA, and a receiver MTA server as an RMTA.

All the outgoing messages of unclassified senders are stored at the SMTAs. For this purpose, an SMTA maintains an outgoing message folder for each *sender*. Instead of a complete message being directly pushed from the SMTA to the RMTA, only the envelope of the message is delivered. In particular, the SMTA notifies the RMTA about the pending message via the new *message identifier* command `MSID` (see Table 1), which contains the unique identifier *msid* of the message. The *msid* is used by the receiver to retrieve the corresponding message.[1] The identifier of a message is generated based on the sender, the receiver, and the message.

3.3 Receiver: Pulling Messages from Unclassified Senders

The new email delivery system grants greater control to receivers regarding if and when receivers want to read messages; senders cannot arbitrarily push messages to them. Receivers can be discriminate about which messages need to be retrieved, and which ones need not. If a receiver indeed wants to read a message, she will inform her own RMTA, and the RMTA will retrieve the message from the SMTA on behalf of the receiver. An RMTA retrieves an email message using the new *get mail* command `GTML` (see Table 1), which includes the identifier *msid* of the message to be retrieved. After the message has been pulled to the RMTA, conventional virus/worm scanning tools and content-based spam filters can be applied to further alert the receiver about potential virus or spam. Therefore, the new email system does not exclude the use of existing email protection schemes. For security reasons, when an SMTA receives the `GTML` command, it needs to

[1] Note the fundamental difference between message pull in the new email system and URL embedded in many current spam messages. The address in the URL is normally not related to the sending machine of the message. In contrast, outgoing messages in the new email system have to be stored on the sender mail servers.

verify that the corresponding message is for the corresponding email receiver, and the requesting MTA is the mail server responsible for the receiver.

3.4 Minimizing the Impact of Intent Messages

It is conceivable that before the majority of spammers are squeezed out of business, a large number of small intent messages may be delivered to Internet email users when DMTP is first deployed on the Internet. A legitimate concern is that email users may be overwhelmed by the number of small intent messages. This problem can be alleviated by, e.g., quarantining intent messages: RMTA will only deliver messages from regular contacts to receivers immediately; all the intent messages from unclassified senders will be first quarantined at the RMTA and only delivered to the receivers periodically in a single message that contains the list of intent messages received. The interval over which the RMTA delivers the list of intent messages to a receiver can be configured by the receiver. A similar idea has been supported in commercial products and employed in real-world systems to handle spam messages [18, 11]. As more spammers run out of business because of the increased adoption of DMTP, intent messages related to spamming will decrease and be less of a concern. (Rather, they will be used for legitimate reasons for first-time correspondents to communicate.)

4 Performance Modeling and Numerical Studies

In this section we first develop a simple mathematical model to investigate the revenue that a spammer can gather by spamming a message to a set of Internet users. Based on this model, we then perform numerical experiments to study the effectiveness of DMTP in controlling spam, and how the behaviors of both spammers and receivers affect the spammers' revenue.

4.1 A Simple Model of Spammer Revenue

Table 2 summarizes the notations used in this section. Consider a spammer s. We assume that s maintains a set of N email addresses to which spam emails can be sent. In this model we establish the expected revenue the spammer can gather by sending a single message to the N email addresses. We assume the spammer owns or rents x machines, each with a unique IP address, to send spam. On average, each machine is capable of sending k messages per unit time. This sending rate

Table 2. Notations used in the spammer revenue model

Notation	Explanation	Setting
N	Number of email addresses maintained by spammer	10M
x	Number of machines used by spammer	62
k	Sending speed of a machine (messages/unit time)	100K
y	Cost paid by spammer per machine per unit time	0.1
g	Gains of spammer for each message delivered	0.005
p	Probability that a receiver reports a spamming machine	0.001
q	Number of reports required for RBL to blacklist a machine	50
r	Mean spam retrieval rate of receivers (retrievals/unit time)	2500

is only constrained by the processing power and Internet access speed of the machines. The spamming task is equally partitioned over the x machines, that is, each machine needs to send the message to N/x receivers. For each machine, the spammer needs to pay y units of cost for each unit of time, e.g., for Internet access or renting machines from hackers or time spent in recruiting zombies. In return, the spammer obtains g units of gain for each message delivered.

For simplicity, we assume there is a central real-time blacklist of well-known spammers, which is used by all receivers. Sender classification is defined at the granularity of IP addresses. Before the spammer starts spamming, we assume that none of the x machines managed by the spammer is listed by the central RBL. Instead, they are in the unclassified-sender class of all N receivers. We assume that intent messages are directly delivered to end users instead of first being quarantined at the RMTAs. When a receiver retrieves a message from the spammer, it will report the IP address of the corresponding SMTA to the central RBL with a probability of p. Furthermore, the central RBL requires at least q reports of a spamming machine before adding the corresponding IP address into its blacklist. After an IP address is added to the blacklist, the spammer can no longer send messages from the corresponding SMTA. To simplify, we assume that the spammer has the precise knowledge of the time when an SMTA is blacklisted and will disconnect the machine to minimize its own cost.

We assume the arrivals of spam retrievals from receivers follow a Poisson distribution, with a mean arrival (i.e., retrieval) rate r retrievals per unit time. Given that the list of email addresses maintained by a spammer is in general large, we assume the spam retrieval rate r is a constant over time. Below we derive the expected revenue $U(t)$ of the spammer at time t, assuming the time for the spammer to start spamming the message to the N receivers is zero.

Let $R(t)$ denote the expected number of receivers who have retrieved the message at time t. It is not too hard to see that $R(t) = \min\{rt, xq/p\}$. Let $f(t)$ denote the expected number of messages delivered by the spammer at time t (across all x machines), we have $f(t) = \min\{N, xkt, R(t)\}$. Consequently, the expected income of the spammer at time t is $gf(t)$. On average, it takes N/r units of time for the spammers to deliver the message to all receivers, and it takes $(q/p)/(r/x)$ units of time for the central RBL to blacklist an SMTA (assuming $r \ll k$ and all x machines are accessed with the same probability). Therefore, the total expected cost $c(t)$ paid by the spammer at time t is $c(t) = xy \min\{t, N/r, (q/p)/(r/x)\}$. Hence, in the DMTP-based email system the total expected revenue of the spammer at time t is

$$U_{DMTP}(t) = gf(t) - c(t) = g \min\{N, xkt, R(t)\} - xy \min\{t, N/r, (q/p)/(r/x)\}. \tag{1}$$

We can similarly derive the total expected revenue of the spammer at time t in the current SMTP-based email system, which is given below

$$U_{SMTP}(t) = g \min\{N, xkt\} - xy \min\{t, (N/x)/k\}. \tag{2}$$

In the above equation, we have assumed that k is large enough that the spammer can finish sending the message to all receivers before the SMTAs are blacklisted.

Comparing Eq. (1) and Eq. (2), we see that while the revenue of the spammer is largely determined by the *sending* speed of its SMTAs in the current SMTP-based email system, in the DMTP-based email system its ability to spam is greatly constrained by the message retrieval behavior of the receivers. The slower the receivers are in retrieving the message, the longer the spammer needs to stay online; the higher the probability is for receivers to report spamming SMTAs to the central RBL, the earlier the spamming SMTAs are blacklisted.

4.2 Numerical Studies

In this section we perform numerical experiments to study the effectiveness of the proposed DMTP protocol in controlling spam using the model developed in the last subsection. We also investigate how the behaviors of both spammers and receivers affect the spammers' revenue. Table 2 (third column) presents the parameter values we used in the numerical studies, unless otherwise stated.

First, we study how the proposed DMTP protocol helps to reduce the maximum revenue of a spammer and forces the spammer to stay online to improve the performance of RBLs. Fig. 3 shows the revenues of the spammer as time evolves in both the current SMTP-based email system (curved marked as *Without DMTP*) and the proposed DMTP-based email system. From the figure we see that, in the current email system, the spammer can gather the maximum revenue (49990) within 2 units of time. This means that the spammer can quickly push out the message to all the receivers and then vanish, long before any RBLs can identify it. In contrast, in the DMTP-based email system, the maximum revenue is 7812 units, only about 16% of the spammer maximum revenue in the current email system. Moreover, in order for the spammer to gather the maximum revenue, the spammer has to stay online for a much longer time window (1240 units of time). This can significantly improve the performance of RBLs. Note also that the revenue will not decrease once it reaches the maximum values. This is because a spammer disconnects an SMTA to minimize the cost once the SMTA has finished sending the message to all receivers in SMTP or it is blacklisted in DMTP.

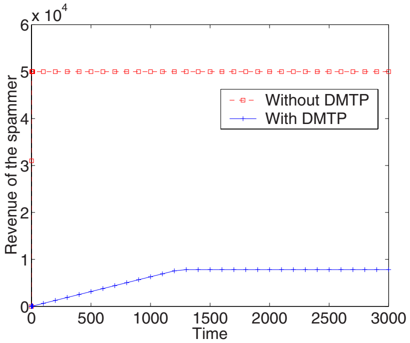


Fig. 3. Expected spammer revenue

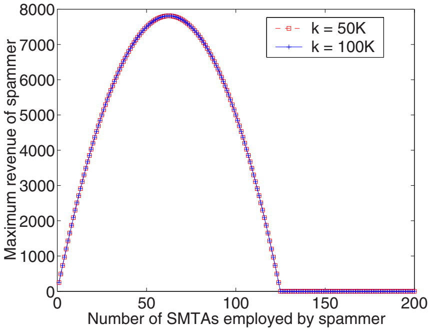


Fig. 4. Impact of number of SMTAs

Next, we investigate the impact of the number of SMTAs employed by a spammer on the maximum spammer revenue in the DMTP-based email system. Fig. 4 shows the *maximum* spammer revenue as a function of the number of SMTAs employed by the spammer for $k = 50K$ and $100K$, respectively. Note first that increasing the sending speed of spam from $k = 50K$ to $100K$ will not result in a higher maximum spammer revenue. Indeed, after the spam sending speed exceeds the spam retrieval rate of receivers, it will not affect the maximum spammer revenue. Now let us examine how the number of SMTAs employed by a spammer will affect the maximum spammer revenue. As we can see that the spammer has some initial gains by increasing the number of SMTAs (when the number is less than 62). This is because as the number of SMTAs increases, it takes a longer time for all the SMTAs to be blacklisted by the central RBL, and the message can be retrieved by more receivers. Fortunately, the spammer cannot indefinitely increase the number of SMTAs to evade RBLs. When the spammer employs more than 62 SMTAs, the maximum revenue actually starts to drop, as the income of delivering the message to new receivers can no longer recompense the cost to deploy the new SMTAs.

In the last set of numerical experiments, we study the effects of the spam retrieval rate of receivers on the maximum spammer revenue. Fig. 5 depicts the *maximum* spammer revenue as a function of the spam retrieval rate of receivers for number of SMTAs $x = 100, 200, 400$, respectively. As we can see from the figure, the maximum spammer revenue decreases as the receivers reduce their retrieval rate of messages from the unclassified SMTAs for all three cases. Moreover, when the retrieval rate is sufficiently low (for example, less than 2000 retrievals per unit time when $x = 100$), the spammer cannot gather any revenue from spamming. More importantly, when a spammer recruits more SMTAs to send spam, it requires a larger threshold of spam retrieval rates for the spammer to gather any revenue (for example, 4000 when $x = 200$, compared to 2000 when $x = 100$). This again demonstrates that spammers cannot gather more revenue by indefinitely recruiting more SMTAs. As more spammers run out of business because of the increased adoption of DMTP, the email spam problem will be effectively controlled on the Internet.

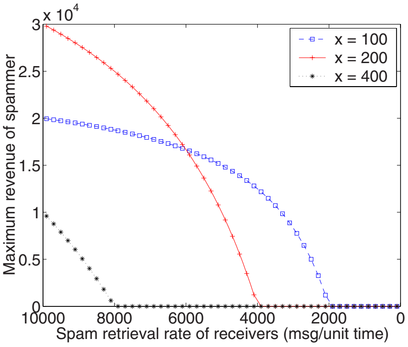


Fig. 5. Impact of spam retrieval rate

5 Implementation and Deployment Issues

In this section we briefly discuss several implementation and deployment issues related to DMTP. We refer interested readers to [4, 6] for details.

Incremental deployment: DMTP can be easily deployed on the Internet incrementally. The basic idea is to combine DMTP with a sender-discouragement scheme (such as asking senders to solve a puzzle). However, unlike existing sender-discouragement schemes, we only require senders in the unclassified-sender class to make the extra effort in sending a message.

Security of message retrieval: A potential concern with the receiver-pull model is security. However as we discuss below, the potential security issue arising from this model is no worse than the current SMTP model. First, important messages are normally communicated amongst regular contacts, which are handled in DMTP in the same way as in the current email system. Secondly, individual users cannot retrieve messages from a remote SMTA directly, they rely on their corresponding RMTAs to retrieve messages (from unclassified senders). Lastly, *msid*s are generated randomly based on the messages (and senders and receivers); they cannot be easily guessed.

Mail forwarding and user-perceived system performance: DMTP does not support open relay mail servers, most of which are blacklisted even today [17]. An organization may deploy multiple mail servers for relaying inbound and outbound mails. Such mail relay servers can be adequately supported by DMTP. We refer the interested readers to [6]. In principle, a message from an unclassified sender is fetched directly from the sender's mail server by the receiver's mail server (or the border mail relay server) in DMTP. Given the ever-increasing network speeds, we do not expect any degradation of user-perceived email reading experience, although some messages–the ones from unclassified senders–need to be retrieved from a remote mail server. We plan to formally study this issue in our future work.

Impact of misbehaved senders on sender mail servers: When sender classification is only supported at the granularity of IP addresses, a single misbehaved sender may destroy the reputation of the sender's corresponding mail server by sending out a large number of spam messages. Fortunately, public mail servers normally establish certain mechanisms to prevent their users from spaming, for example, by imposing a quota on the number of messages a user can send everyday. In general, it is hard for spammers to send spam messages through public mail servers. We will further investigate this problem in our future work.

6 Related Work

The most widely deployed anti-spam solutions today are reactive content filters that scan the contents of the message at the receiver's MTA after the message has been delivered. However, none of them can achieve 100% accuracy, and spammers quickly adapt to counter the strategies used by these filters. In addition, content filtering will no longer serve as long-term viable solution once email messages begin to be encrypted using receivers' public keys [16]. Instead, we have advocated fundamental changes in protocol-level design to a pull-based model.

Like DMTP, FairUCE [2] also advocates the usage of sender classifiers. However, it is still a push-based model in which network reputation, along with receiver defined whitelist and blacklist, is used to determine whether to accept a message. IM2000 [1] also advocates a pull-based model like DMTP. However, unlike DMTP, all outgoing messages need to be stored at sender MTAs and receivers need to retrieve all the messages remotely, regardless of where the messages come from. In addition, IM2000 is not incrementally deployable and requires massive infrastructure changes. Li *et al* proposed a method to slow down spam delivery by damping the corresponding TCP sessions [15]. However, the long-term impact of modifying the behavior of TCP for a specific application is not clear, and spammers may respond by changing sender MTA's TCP behavior. In the Greylisting [10] approach, a message from a new sender is temporarily rejected upon the first delivery attempt, the underlying assumption being that spammers will not re-send a message whereas regular MTAs will. However, it is only a matter of time before spammers adapt to this technique by re-sending their message. Sender authentication schemes such as [3, 14] can help improve the accountability of email senders. However, they cannot control the delivery of spam by themselves.

7 Conclusion and Ongoing Work

In this paper we examined the architectural aspects of the current email system that are responsible for the proliferation of spam, and proposed a Differentiated Mail Transfer Protocol to control spam. In addition, we also developed a formal model to study the performance of DMTP. Through numerical experiments we demonstrated that DMTP can significantly reduce the maximum spammer revenue. Moreover, it also forces spammers to stay online for longer periods of time, which helps improve the performance of real-time blacklists of spammers. Currently we are developing a prototype of DMTP. We plan to further investigate the performance and other potential design issues of DMTP based on the prototype and simulations.

References

1. D. Bernstein. Internet mail 2000 (IM2000). http://cr.yp.to/im2000.html.
2. IBM Corporation. Fair use of unsolicited commercial email FairUCE, November 2004. http://www.alphaworks.ibm.com/tech/fairuce.
3. M. Delany. Domain-based email authentication using public-keys avertised in the DNS (domainkeys). Internet Draft, August 2004. Work in Progress.
4. Z. Duan, Y. Dong, and K. Gopalan. DMTP: Controlling spam through message delivery differentiation. Technical Report TR-041025, Department of Computer Science, Florida State University, October 2004.
5. Z. Duan, K. Gopalan, and Y. Dong. Push vs. pull: Implications of protocol design on controlling unwanted traffic. In *Proc. USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI 2005)*, July 2005.

6. Z. Duan, K. Gopalan, and Y. Dong. Receiver-driven extensions to SMTP. Internet Draft, January 2006. Work in Progress.
7. Ferris Research. Spam control research reports. http://www.ferris.com/.
8. J. Goodman and R. Rounthwaite. Stopping outgoing spam. In *Proc. of EC'04*, 2004.
9. P. Graham. A plan for spam. *http://www.paulgraham.com/spam.html*, January 2003.
10. E. Harris. The next step in the spam control war: Greylisting. White Paper, August 2003.
11. ITS. Spam at the university of hawaii. http://www.hawaii.edu/infotech/spam/spam.html. Last checked: 11/19/2005.
12. A. Juels and J. Brainard. Client puzzles: A cryptographic defense against connection depletion attacks. In *Proceedings of NDSS-1999*, February 1999.
13. J. Klensin. Simple mail transfer protocol. RFC 2821, April 2001.
14. M. Lentczner and M. W. Wong. Sender policy framework (spf): Authorizing use of domains in MAIL FROM. Internet Draft, October 2004. Work in Progress.
15. K. Li, C. Pu, and M. Ahamad. Resisting spam delivery by TCP damping. In *Proceedings of First Conference on Email and Anti-Spam (CEAS)*, July 2004.
16. P. Mannion. Interview: Ethernet's inventor sounds off. *Information Week*, November 2005.
17. RBL. Real-time spam black lists (rbl). http://www.email-policy.com/Spam-black-lists.htm.
18. Sophos Plc. Sophos plc. http://www.sophos.com/.
19. The Editors. Product of the year: Spam? *Information Week*, January 2004.

Delay Performance Analysis for an Agile All-Photonic Star Network

Cheng Peng, Peng He, Gregor v. Bochmann, and Trevor J. Hall

Centre for Research in Photonics,
School of Information Technology and Engineering,
University of Ottawa, Ottawa, ON, K1N 6N5, Canada
{cpeng, penghe, bochmann, thall}@site.uottawa.ca

Abstract. In this paper, we study the delay performance of a centrally-controlled agile all-photonic star WDM network that provides multiplexing in the time domain over each wavelength. We consider two timeslot allocation strategies, First-Fit (FF) and First-Fit+Random (FFR), as well as network scenarios with different propagation delays. Both theoretical analyses and simulation experiments are conducted to evaluate the delay performance of the network. Through analytical and simulation results, we show that allocating residual free bandwidth can significantly improve queuing delay performance under light traffic load while maintaining good delay performance under heavy traffic load, especially for a network scenario with large propagation delays. The results obtained can be used to guide the design of scheduling algorithms especially for large-scale networks.

1 Introduction

The term "agility" in optical networks describes the ability to deploy bandwidth on demand at fine granularity, which radically increases network efficiency and brings to the user much higher performance at reduced cost. One possible scheme to provide such agility in WDM networks is multiplexing in the time domain, which is based on the principle of Time Division Multiplexing (TDM)[1]. In such a context, optical switches along lightpaths must be scheduled to reconfigure every timeslot, or every few timeslots, for bandwidth sharing. The centrally-controlled Agile All-Photonic Networks (AAPN)[1][2] can provide such agility.

As shown in Figure 1, an AAPN consists of a number of hybrid photonic/electronic edge nodes connected together via a core node that contains a stack of bufferless transparent photonic space switches one for each wavelength. A scheduler at a core node is used to dynamically allocate timeslots over the various wavelengths to each edge node. An edge node contains a separate buffer for the traffic destined to each of the other edge nodes. These buffers are called Virtual Output Queues (VOQs) [3] and are used to eliminate the Head-Of-Line blocking problem associated with First-In-First-Out (FIFO) queuing [4]. Traffic aggregation is performed in these buffers, where packets are collected together in fixed-size slots (or, alternatively, bursts) that are then transmitted as single units across the network via optical links. At the destination edge

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 368 – 378, 2006.

node the slots are partitioned, with reassembly as necessary, into the original packets. The optical core network does not provide wavelength conversion or buffering, which provides major network architecture simplifications and hardware reductions.

The focus of this paper is on the delay performance of the AAPN with two timeslot allocation strategies as well as network scenarios with different propagation delays. We first review related work in the literature and introduce the signaling protocols between core and edge nodes. Then we provide both theoretical analyses and simulation experiments to evaluate the delay performance of the network by applying these two strategies to AAPN. In the last part of the paper, we conclude that allocating residual free bandwidth can significantly improve queuing delay performance under light traffic load while maintaining good delay performance under heavy traffic load.

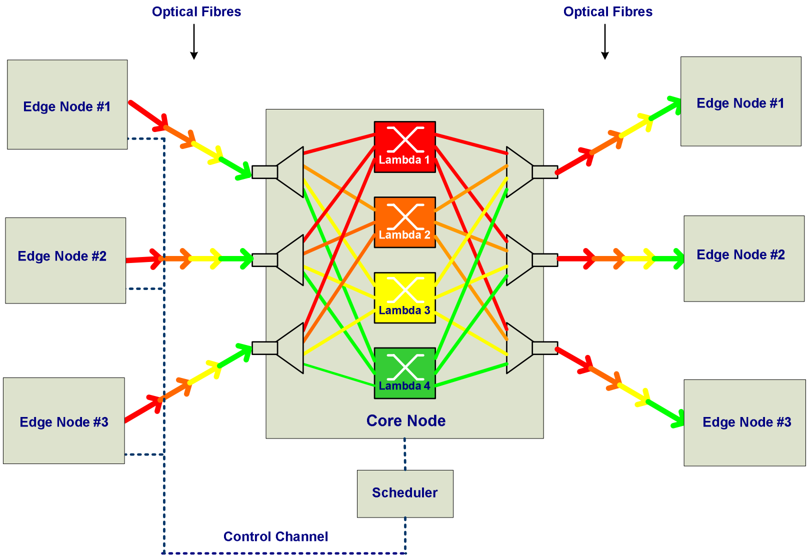


Fig. 1. AAPN star topology

2 Related Work

In the literature, the delay performance of different scheduling algorithms for an Input Queued (IQ) switch (a switching fabric equipped with buffers at its input ports) have been studied extensively (e.g. [5][6]). The work is relevant to AAPN since the star network formed by each wavelength space switch with its attached edge nodes can be viewed as a distributed IQ switch. In fact, the AAPN architecture may be viewed as a distributed three-stage Clos packet switch: the edge nodes can be logically split in two parts (the source and the destination modules) and the core node may be explicitly drawn with its de-multiplexers/multiplexers and its wavelength space switches in parallel. The connections between the respective source/destination edge nodes and the core node are seen now as unidirectional (shown in Figure 1).

The delay performance analyses for an IQ switch usually shown in the literature cannot be compared to AAPN because it implies zero propagation delay between the edge nodes (input buffers) and the core node (switch fabric). The propagation delay, however, cannot be ignored since AAPN can be deployed in the backbone of national or large metropolitan networks.

The performance of passive star optical networks was studied extensively in [7][8][9]. By “passive” it is meant that the core node uses couplers or Array Waveguide Grating (AWG) optical devices. Though the network topology is the same as AAPN the switching mechanisms applied to the core node are quite different since passive AWG optical devices are used, as opposed to the dynamic photonic switch fabrics used in the AAPN core node. A scheduling algorithm based on timeslot allocation was proposed in [9] but the delay performance was not studied. The scheduling algorithms discussed in [7] and [8] are quite different from our work because they do not multiplex slots in the time domain over each wavelength.

3 Signaling and Scheduling Strategies in a TDM-AAPN

3.1 Signaling Protocols

When the AAPN operates in TDM mode (TDM-AAPN), each edge node signals a bandwidth request (estimated mainly from queue state information) to the core along control channels (shown as dotted lines in Figure 1) before sending the slots. The scheduler at the core allocates timeslots to each edge node over an appropriate wavelength based on the request. The schedule is signaled back to inform each edge node of the timeslots that it may use to transmit its traffic for each destination, and the core wavelength switches are re-configured in coordination with the edge nodes according to the bandwidth allocated.

3.2 Scheduling Strategies

The main task of a scheduler at a core node is to allocate timeslots over an appropriate wavelength to edge nodes.

An ideal scheduler may allocate timeslots in such a way that each edge node may be granted the earliest possible available timeslots based on its bandwidth request. By “earliest possible” we mean the earliest timeslot that an edge node can send a slot without contention. In order to simulate the ideal bandwidth allocation mechanism, a scheduling strategy, called *First-Fit* (FF), is studied, where the earliest possible timeslots can always be allocated without blocking. The timeslots granted by the scheduler in response to the request are called *reserved* timeslots for the slot for which the request was issued.

Another scheduling strategy is called *First-Fit+Random* (FFR) where the scheduler allocates the earliest possible available timeslot for each edge node based on its bandwidth request and randomly allocates residual free timeslots over all the output links of the AAPN core node. Such randomly allocated timeslots are referred to as *unreserved*. The benefits of this strategy are that slots may be transferred without waiting for their reserved timeslots if an unreserved timeslot assigned to the correct destination is available earlier, which reduces the queuing delay of the slots. Evidently, a slot cannot be sent later than its reserved timeslot.

4 Analytical Analyses of Delay Performance

In this section, analytical analyses of the delay performance for the two timeslot allocation strategies, FF and FFR, over a single wavelength are discussed in the context of AAPN.

4.1 Assumptions

We assume that an AAPN core contains a $N \times N$ photonic space switch for a single wavelength. Each source edge node maintains a VOQ with a FIFO queuing policy for each destination edge node. The capacity of these VOQs is assumed to be infinite. It is also assumed that, in every timeslot, there is an independent and identical probability ρ (load) that traffic arrives at an edge node. The arrivals therefore follow a Poisson distribution. We assume that traffic is equally likely destined for each edge node. The longest propagation delay between core and edge nodes is assumed to be d .

4.2 Analytical Analyses of Delay Performance

Two factors affect the queuing delay performance. One is the scheduling delay Δ; the other is the signaling delay $d_{signaling}$. The scheduling delay describes the waiting time introduced by the scheduler to resolve contention. A bound for Δ follows Shah's delay model in [6] which is slightly tighter than Leonardi's [5] for uniform traffic. According to Section 3, the signaling delay is defined as the round-trip time (measured in timeslots) between the core and the edge nodes.

$$d_{signaling} = 2d \tag{1}$$

4.2.1 FF Strategy

The waiting time of a slot that is transmitted through its reserved timeslot is denoted by D_{rsv}. The D_{rsv} is determined by the sum of the signaling and scheduling delay, that is:

$$D_{rsv} = d_{signaling} + \Delta \tag{2}$$

The signaling delay $d_{signaling}$ can be calculated according to (1). The scheduling delay Δ can be calculated according to following arguments.

Given a $N \times N$ photonic space switch, the probability P_{dest} that a slot from a given source edge node is destined to a given destination edge node is:

$$P_{dest} = \frac{1}{N} \tag{3}$$

The probability ρ_{dest} that a slot arrives at a given source edge node and is destined to a given destination edge node is

$$\rho_{dest} = \rho \times P_{dest} = \frac{\rho}{N} \tag{4}$$

In the context of AAPN, there are a total of N VOQs (one for each edge node) with slots destined to the same edge node that contend for the output link of the core node, and there are a total of N VOQs (in one edge node) that contend for the input link of the core node. Therefore, the link load ρ_{link} is:

$$\rho_{link} = N \cdot \rho_{dest} = \rho \tag{5}$$

The scheduling delay that a slot waits in a VOQ can be approximated [6] as,

$$\Delta = \frac{N-1}{N} \cdot \frac{\rho_{link}}{1-\rho_{link}} \cdot \frac{1}{\rho_{link}/N} = \frac{N-1}{1-\rho} \tag{6}$$

Introducing (1) and (6) to (2), we have

$$D_{rsv} = 2d + \frac{N-1}{1-\rho} \tag{7}$$

For FF strategy, a slot at an edge node can only be sent out through its reserved timeslot. Hence the queuing delay of FF strategy D_{FF} is:

$$D_{FF} = D_{rsv} = 2d + \frac{N-1}{1-\rho} \tag{8}$$

4.2.2 FFR Strategy

For FFR strategy, it is not necessary for a slot to wait D_{rsv} timeslots for transmission because the slot may be sent out earlier by using an unreserved timeslot. We denote by D_{ursv} the waiting time of a slot that is transmitted through an unreserved timeslot. The D_{ursv} is determined by the scheduling delay only, i.e., when a slot reaches the head of its VOQ, it will depart on the first unreserved timeslot if one is available before its reserved timeslot; otherwise it waits for its reserved timeslot. Hence,

$$D_{ursv} = \Delta \tag{9}$$

For the AAPN deployed in the backbone of national or large metropolitan networks (optical links are more than 100km), nearly all slots are transmitted through unreserved timeslots if the load is less than 0.5 due to the facts that (1) the number of the waiting slots is on average less than that of the unreserved timeslots and (2) the time that a slot waiting for its reserved timeslot is rather long. In case the load exceeds 0.5, it means that some slots are forced to be transmitted through their reserved timeslots because of the shortage of the unreserved timeslots. Based on the thoughts, we discuss the expected queuing delay of the FFR in two scenarios.

1. The expected queuing delay of the FFR when $\rho \geq 0.5$

The probability P_{su} that a slot is sent out through an unreserved timeslot is determined by three conditions: (1) the timeslot is unreserved; (2) the timeslot is destined to the same edge node as the slot; (3) the corresponding VOQ is not empty.

Let us consider a timeslot allocation $A(i)$ by a FFR scheduler for a given edge i, where $i \in \{0,1,...N-1\}$.

Denoted by P_{ursv}, the probability that $A(i)$ is an unreserved timeslot departing from edge node i is:

$$P_{ursv} = 1 - \rho_{link} \tag{10}$$

The probability that $A(i)$ is destined to a given edge j is denoted by P_j where $j \in \{0,1,...N-1\}$. According to the uniform traffic assumption, the reserved timeslots are equally likely destined for each edge node. Due to the random allocation for the residual bandwidth, the unreserved timeslots are equally likely destined for each edge node as well. Thus, for any $j \in \{0,1,...N-1\}$, we have

$$P_j = \frac{1}{N} \tag{11}$$

The probability that the j th VOQ is not empty is denoted by P_{VOQ_j} for any $j \in \{0,1,...N-1\}$. In the context of AAPN, there are a total of N VOQs (one for each edge node) with slots destined to the same edge node that contend for the output link of the core node, and there are a total of N VOQs (in one edge node) that contend for the input link of the core node. Hence,

$$P_{VOQ_j} = \rho_{link} \tag{12}$$

By (5), (10), (11) and (12), we have

$$P_{su} = P_{ursv} \cdot P_j \cdot P_{VOQ_j} = \frac{\rho(1-\rho)}{N} \tag{13}$$

Accordingly, the probability P_{sr} that a slot is sent out through its reserved timeslot is

$$P_{sr} = (1 - P_{su}) = 1 - \frac{\rho(1-\rho)}{N} \tag{14}$$

Consequently, the expected queuing delay $D_{\geq 0.5}$ for $\rho \geq 0.5$ can be calculated by the following equation.

$$D_{\geq 0.5} = P_{sr} D_{rsv} + P_{su} D_{ursv} \tag{15}$$

Introducing (1), (2), (7), (9), (13) and (14) to (15), we have

$$D_{\geq 0.5} = 2d(1 - \frac{\rho(1-\rho)}{N}) + \frac{N-1}{1-\rho} \tag{16}$$

2. The expected queuing delay of the FFR when $\rho < 0.5$

For $\rho < 0.5$, more unreserved than reserved timeslots are allocated by the FFR. Almost all slots can be sent out through unreserved timeslots. Based on these thoughts, we assume for now that all slots are sent out through unreserved timeslots, which is equivalent to the case that the core node randomly allocates unreserved timeslots without knowing the bandwidth requests and signals the allocations back to the edge nodes.

According to the assumption, the equivalent effective bandwidth for each link degrades to $1-\rho$ of the link bandwidth since we assume no slots are transmitted through reserved timeslots. Hence, the equivalent link load $\rho_{eqv-link}$ increases up to

$$\rho_{eqv-link} = \frac{\rho}{1-\rho} \tag{17}$$

The scheduling delay Δ that a slot waits in a VOQ can be approximated [6] as

$$\Delta = \frac{N-1}{N} \cdot \frac{\rho_{eqv-link}}{1-\rho_{eqv-link}} \cdot \frac{1}{\rho_{eqv-link}/N} = \frac{N-1}{1-\rho_{eqv-link}} = \frac{N-1}{1-2\rho}(1-\rho) \tag{18}$$

Consequently, the expected queuing delay $D_{<0.5}$ for $\rho < 0.5$ is

$$D_{<0.5} = \Delta = \frac{N-1}{1-2\rho}(1-\rho) \tag{19}$$

3. The expected queuing delay of the FFR

For (19), given a N, we have

$$\lim_{\rho \to 0.5} D_{<0.5} = \infty \tag{20}$$

Equation (20) indicates that the delay will become infinite when the load approaches to 0.5, which is unrealistic. The reason of this result is the assumption that all slots are sent out through unreserved timeslots when $\rho < 0.5$. Actually, when the load increases, some slots have to use their reserved timeslots so the queuing delay would be bounded by Equation (16). Consequently, the expected queuing delay of the FFR D_{FFR} can be expressed by the following equation.

$$D_{FFR} = \begin{cases} D_{\geq 0.5} & \rho \geq 0.5 \\ \min(D_{<0.5}, D_{\geq 0.5}) & otherwise \end{cases} \tag{21}$$

Note that Equation (21) can be solved by combining (16) and (19).

5 Simulation Results and Discussions

In this section, we present both the analytical and simulation results of the delay performance of AAPN using the two scheduling strategies, FF and FFR, with different propagation delays. We demonstrate the accuracy of our analytical techniques by comparing analytical results to simulation. In the simulations, the traffic requests arrive at the network following a Poisson process, and the holding time is exponentially distributed. We assume that all the source-destination node pairs have the same traffic load. The duration of a timeslot is 10 microseconds. The simulation lasts 1,000,000 timeslots.

There are three factors that affect the queuing delay under a given load, i.e., the scheduling strategies (whether using unreserved timeslots or not), the scalability of the AAPN core node (measured in the port dimension N), and the longest propagation delay between the core and the edge nodes, d (measured in timeslot). We study how these factors affect the delay performance in this section.

In Section 4, we assume that all slots are sent out through unreserved timeslots when $\rho < 0.5$. With this assumption, we analyze the delay performance of the FFR. Figure 3 shows the simulation result of the probability (P_{sr} in Equation (14)) that a slot is transmitted through its reserved timeslot under the FFR scheduling strategy. As we can see, when the load is less than 0.5 (light load), the probability is almost zero and thus can be ignored. When the load becomes larger than 0.5, the probability increases very fast for all three propagation delays and hence cannot be ignored. This simulation shows that the assumption for $\rho < 0.5$ is correct.

In Figure 4 (Equation (8) and (21)), three curves are presented. The solid ones show the expected queuing delay of the FFR, where some slots are sent out earlier by using

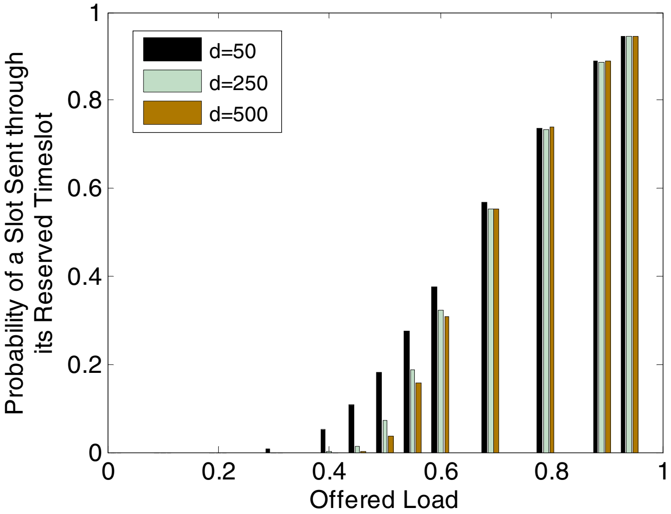


Fig. 3. The probability of a slot transmitted through its reserved timeslot. (N=8)

unreserved timeslots. The simulation results of the FFR are shown as point marks (no lines). The dotted lines, as a benchmark, give the expected queuing delay by applying the FF strategy, where all slots are sent out through their reserved timeslots.

It is shown that the delay curves of the FFR (Equation (21)) are matched well with the simulation curves, both in amplitude and in the trend of curves in the low-load and high-load regions, which confirms the correctness and exactness of our analytical analysis proposed in Section 4 in general. However, as mentioned earlier, in the region of medium load, the analytical model is not precise because of the assumptions made.

As shown in Figure 4, the delay curves can be divided into three regions according to load, i.e., $\rho < 0.5$, $\rho \approx 0.5$ and $\rho > 0.5$.

In the first region ($0 \le \rho < 0.5$), where the load is light, both FF and FFR curves are fairly "flat". The FF curves are just above $2d$ because of the signaling delay that is equal to $2d$. Compared to the scheduling delay, the signaling delay of the FF dominates the queuing delay in the low load region. As shown in Figure 4, the FFR curves are just above zero. Apparently, FFR outperforms FF greatly because nearly all the slots can be transported by unreserved timeslots.

The scale of such an improvement is different for different propagation delays. It is shown that FFR can contribute a significant improvement to the delay performance compared to FF in an AAPN with large propagation delays. The reason is that the propagation delay has nearly no influence on the delay performance for FFR (Equation (19)) when $\rho < 0.5$. It hence can be concluded that allocating residual free

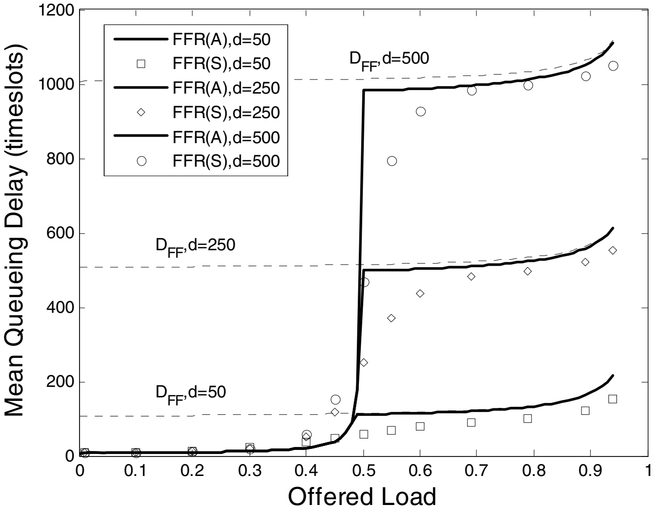


Fig. 4. Queuing delay as a function of offered load for AAPN under various scheduling strategies and different propagation delay (N=8)

bandwidth can significantly improve queuing delay performance in the light load region, especially when the propagation delay is large, e.g., for a WAN.

In the second region ($\rho \approx 0.5$), the FFR curves dramatically rise and approach the FF curves. This can be explained by Equation (20) that is the slots are largely accumulated in the VOQs so that some of them are forced to use their reserved timeslots for transmission. It is observed that the simulated FFR curves are not as sharp as the analytical ones. This is because the analytical model assumes that all slots are sent out through unreserved timeslots when $\rho < 0.5$, which ignores the transition by which slots gradually use more reserved timeslots as ρ increases in the first region.

In the third region ($0.5 < \rho < 1$), where the load is heavy, slots are more likely to be sent through their reserved timeslots and hence experience both the signaling delay and the scheduling delay. In this region, signaling delay dominates the delay performance. Thus the FFR has only a limited improvement to the delay performance of the FF. It can be concluded that allocating residual free bandwidth gives a small improvement to the queuing delay performance in the heavy load region.

6 Conclusions and Remarks

In this paper, we study the delay performance of an agile all-photonic star network with centralized schedulers working in TDM mode. A scheduling strategy, called *First-Fit* (FF), which emulates an ideal bandwidth allocation that allocates the earliest available timeslot for each edge node based on its bandwidth request, is studied. The FF shows long queuing delay especially for large-scale networks even if the traffic load is light. Another scheduling strategy, called *First-Fit+Random* (FFR), is studied and shows a significant improvement of the queuing delay performance in the light load region. Through analytical and simulation results, we show that allocating residual free bandwidth can significantly improve the queuing delay performance under light traffic load while maintaining good delay performance under heavy traffic load, especially for a network scenario with large propagation delays.

The model proposed in the paper is for the AAPN deployed in the backbone of national or large metropolitan networks where the propagation delay is quite large. The accuracy of the model may decrease in the LAN environment due to the assumption that all slots are sent out through unreserved timeslots when $\rho < 0.5$. Figure 3 shows that more reserved timeslots are used for transmission in the region of $\rho < 0.5$ with a smaller propagation delay. It therefore reminds us that the propagation delay in LAN can be ignored since the scheduling delay Δ dominates the delay performance which can thus be modeled by classic queuing theories.

The conclusion in this paper can be used to design scheduling algorithms. For example, a Birkhoff-von Neumann decomposition based timeslot allocation algorithm is discussed in [10] where the residual bandwidth is allocated in a round-robin way to gain good delay performance. Furthermore, the residual bandwidth can also be allocated randomly with a weight proportional to the average traffic to the particular destination (average over some recent time period) to adapt to non-uniform traffic.

The current FFR reserves bandwidth for incoming traffic and tries to take advantage of unreserved timeslots. The reserved timeslots can only be used to send the slots that reserve them even if they have been sent out through the unreserved timeslots, which implies that these timeslots cannot be dedicated to others in this case. Hence, one of our future works is to study an enhanced version of FFR, in which a reserved timeslot can become unreserved if its associated slot has already been send out.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and industrial and government partners, through the Agile All-Photonic Networks (AAPN) Research Network. Dr. Trevor J. Hall holds a Canada Research Chair in Photonic Network Technology at the University of Ottawa and is grateful to the Canada Research Chairs Program for their support. The authors are also indebted to Dr. Sofia Paredes and Dr. Jun Zheng for their insightful comments and careful reading of the manuscript.

References

1. G. v. Bochmann, M.J. Coates, T. Hall, L. Mason, R. Vickers and O. Yang, “The Agile All-Photonic Network: An architectural outline”, Proc. Queen's University Biennial Symposium on Communications, 2004, pp.217-218
2. L.G. Mason, A. Vinokurov, N. Zhao and D. Plant, "Topological Design and Dimensioning of Agile All Photonic Networks", accepted to "Computer Networks", special issue on Optical Networking, edited by Prof. Harry Perros
3. Y. Tamir and G. Frazier, “High performance multiqueue buffers for VLSI communication switches”, In Proceedings of 15th International Symposium on Computer Architecture (ISCA), May/June 1988, pp. 343-354
4. N. McKeown, “The iSLIP Scheduling Algorithm for Input-Queued Switches”, IEEE/ACM Transactions On Networking, vol. 7, April 1999, pp.188-201
5. E. Leonardi, M. Mellia, F. Neri, and M.A. Marsan, “Bounds on delays and queue size averages and variances in input-queued cell-based switches”, Proc. of the IEEE INFOCOM, Anchorage, USA, April 2001,pp.1095-1103
6. D. Shah, and M. Kopikare, “Delay bounds for approximate Maximum weight matching algorithms for input-queued switches”, Proc. of the IEEE INFOCOM, New York, USA, June 2002, pp. 1024-1031
7. M. Maier, M. Scheutzow, M. Reisslein, and A. Wolisz, "Wavelength Reuse for Efficient Transport of Variable-Size Packets in a Metro WDM Network," in Proc., IEEE INFOCOM, vol. 3, June 2002, pp. 1432-1441
8. N. Kamiyama, "A Large-Scale AWG-Based Single-Hop WDM Network Using Couplers With Collision Avoidance," IEEE/OSA JLT, vol. 23, no. 7, July 2005, pp. 2194-2205
9. A. Bianco, E. Leonardi, M. Mellia, and F. Neri, "Network Controller Design for SONATA - A Large-Scale All-Optical Passive Network," IEEE JSAC, vol. 18, no. 10, Oct. 2000, pp. 2017-2028
10. C. Peng, G. v. Bochmann and T. J. Hall, “Quick Birkhoff-von Neumann Decomposition Algorithm for Agile All-Photonic Network Cores”, accepted by 2006 IEEE International Conference on Communications (ICC 2006), Istanbul, Turkey, June 2006

Designing Scalable WDM Optical Interconnects Using Predefined Wavelength Conversion

Haitham S. Hamza and Jitender S. Deogun

Department of Computer Science & Engineering,
University of Nebraska-Lincoln,
Lincoln, NE 68588-0115, USA
{hhamza, deogun}@cse.unl.edu

Abstract. This paper investigates the problem of designing scalable and cost-effective wavelength division multiplexing (WDM) optical interconnects. We propose a new design for WDM optical interconnect that has several advantages over existing designs. First, wavelength conversion occurs between two *predefined* wavelengths. This not only eliminates the need for expensive wide-range wavelength converters, but also ensures high scalability as the conversion range is independent of the number of the wavelengths in the system. Second, the new design requires a smaller number of switching elements compared to most of the recent best interconnect designs.

1 Introduction

A Wavelength Division Multiplexing (WDM) interconnect provides the basic functionality of switching a signal arriving at an input fiber on one of the possible wavelengths to an output fiber on a possibly different wavelength, while maintaining the signal in the optical domain. In the absence of *Wavelength Converters* (WCs); an optical interconnect can only switch signals in the *"space domain"*: an input signal on a given wavelength can be connected to any output fiber without changing its wavelength (given that this wavelength is free on the required output fiber). Switching only in the space domain, however, limits the capability of the interconnect. Thus, introduction of switching in both space and wavelength domains enhances the capability of interconnects. Several interconnect with space and wavelength switching capabilities have been proposed over the last few years, e.g. [11] [17] [38] [39].

However, as WDM enables more and more wavelengths per fiber, the design of cost-effective and scalable WDM optical interconnects becomes a real challenge as the number of required switches and WCs increases. Large number of switches and WCs can lead to architectures that are impractical or economically infeasible. In general, the cost of a WDM interconnect is dominated by two factors: the *switching* cost and the *wavelength conversion* cost [11][38] [39]. Switching cost is proportional to the total number of switching elements (SEs) in the interconnect [38][39].

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 379–390, 2006.

Conversion cost, on the other hand, depends on the total number of WCs as well as the conversion cost of each WC. Let Λ be the set of wavelengths in the network, and let S and D be two sets of wavelengths such that $S \subseteq \Lambda$ and $D \subseteq \Lambda$. Denote by WC(S, D) a WC that is capable of converting any wavelength in set S to any wavelength in set D. The conversion cost of WC(S, D) is therefore proportional to $|S|.|D|$, $1 \leq |S|, |D| \leq |\Lambda|$ [38]. Obviously, the cheapest WC is the one with $|S| = |D| = 1$, or a *Fixed-range* WC. Most existing WDM interconnects, however, make use of *Full-range* WCs (FWC) where $|S| = |D| = W$ [11][39]. Typical WDM interconnects are large, and for these FWCs can be very expensive and difficult to implement and thus, more recent interconnects adapt *Limited-range* WCs (LWCs) instead of FWCs [38].

Using LWCs, however, may not lead to cost-effective and scalable WDM design, particularly for large interconnects. This because, the conversion range of LWCs is proportional to the number wavelengths in the system. Therefore, as the number of wavelengths in the system increases, so does the conversion range of LWCs. We argue that, in order to develop cost-effective and highly scalable WDM interconnects, *the range of used wavelength conversion must be independent of the number of wavelengths in the system.* Accordingly, in this paper, we investigate a new design for WDM optical interconnects that require *"only"* wavelength conversion between two predefined wavelengths while providing a full-connectivity between input and output fibers. The new design exploits the potential of the *Wavelength Exchange Optical Crossbar* (WOC) — a device that can switch signals *simultaneously* and *seamlessly* both in space and wavelength domains [17]. The proposed design follows the recursive structure of the well-known Clos network [5], and hence, scalability occurs in an orderly fashion.

The reminder of the paper is organized as following. Section 2 provides an overview of existing WDM optical interconnect design approaches. The proposed design is presented in Section 3. Section 4 investigates the hardware complexity of the new design. A comparison of different interconnect designs is given in Section 5; Conclusions are presented in Section 6.

2 Existing Designs for WDM Optical Interconnect

Several WDM interconnect designs have been investigated in the literature, e.g. [1], [3], [9], [11], [13]–[16],[20], [26], [27], [36]–[38]. A generic $FW \times FW$ WDM optical interconnect (where F and W represent, respectively, the number of fibers and wavelengths per fiber) consists of two main modules: a *space switching* unit (a switching unit, for short) and one or more *wavelength conversion* units (conversion units). In the following, we classify existing interconnect architectures based on the design of their switching and conversion units.

- Design of Switching Units. Switching units can be broadly classified into two main categorizes: *Single Stage* and *Multistage* designs:
  1. *Single Stage:* In this design, a single $FW \times FW$ switching fabric is used [37], [38]. Such a design requires F^2W^2 switching elements which can be

very expensive for a typical large-scale WDM interconnect with hundreds of fibers and wavelengths.

2. *Multistage: Multistage Interconnection Networks* (MINs) are used to economically realize large-scale interconnects [32]. A MIN interconnects a set of input ports to a set of outputs ports using several stages of fixed-size switching modules. Electronic MINs have been extensively investigated in the literature and several designs were propagated to the optical domain, e.g. [1], [3], [9], [13], [14], [17], [20], [26], [27], [38], [39].

- Design of Conversion Units. Conversion units are realized using wavelength converters. Regardless of their technology, WC can be generally classified into *Full-range* or *Limited-range* converters. Designs for conversion units can be classified into three main categorizes:
  1. *Dedicated Wavelength Conversion:* In this approach, each input and/or output port is assigned a dedicated WC. Thus, a $FW \times FW$ interconnect under this design will have *at least* FW WCs. These WCs can be FWCs, e.g. [11], [37], [39], or LWCs (and SWCs), e.g. [15], [16], [36], [38]. With large number of wavelengths, however, this approach can lead to designs that are impractical, especially if FWCs are used.
  2. *Wavelength Converter Banks:* In this design scheme, a pool of WCs is allocated in the interconnect [23], [28]. Only signals that require wavelength conversion are directed to this pool and then switched to the required output port. In such a design, there is a trade-off between the number of converter banks and WCs within each bank; the cost and complexity of switching; and the permutation capacity of the interconnect (i.e. the number of permutations patterns that the interconnect can realize).
  3. *Bulk Wavelength Conversion:* This scheme adapts *bulk* WCs that are capable of converting multiple wavelengths *simultaneously.* Interconnects that use bulk WCs are known as *wave-mixing* interconnects [1], [27]. Wave-mixing interconnects considerably reduce the number of WCs compared to other design approaches, while avoiding the added complexity of the shared conversion bank design discussed above. However, these designs may introduce up to $O(logW)$ wavelength conversion stages [27] , and hence, the length of the signal path within the switch increases. These extra stages not only increase hardware complexity but also increase the length of the signal path. As a result, delay, signal attenuation, and accumulated cross-talk noise are also increased [8].

3 The Proposed WDM Interconnect

The proposed design is based on the well-known Clos network [5]. An $N \times N$ 3-stage Clos network, denoted as $C(m, n, r)$, has r switches of $n \times m$ size each in the first stage; m switches of $r \times r$ size each in the second; and r switches of $m \times n$ size each in the third stage, and $N = r.n$. The parameter m determines the blocking characteristics of the Clos network. For brevity, we consider only rearrangable nonblocking interconnects, i.e. $m = n$ [22], and extensions to strictly nonblocking designs follows easily. In the following, we first review the main building blocks of the proposed interconnect and then we present the structure of the new design.

3.1 Wavelength Exchange Optical Crossbar

This section briefly reviews the concepts of WOCs and WDM crossbar switches used in the proposed design. A WOC has two input ports, two output ports, and a control signal (Figure 1) [17]. The input to a WOC is two signals S_1 at wavelength λ_a, and S_2 at wavelength λ_b. When the control signal is *OFF*; an input signal to the WOC appears at an output port with the same wavelength. Conversely, when the control signal is *ON*, the WOC performs both switching and conversion simultaneously. WOC can be realized by simultaneous power exchange between the two input signals, a phenomenon that has been theoretically and experimentally demonstrated using *Four Wave Mixing (FWM)* [24][25], and *Photonic Crystal* [2]. It is worth noting that, a WOC performs *predefined fixed* wavelength conversion and hence $|S| = |D| = 1$.

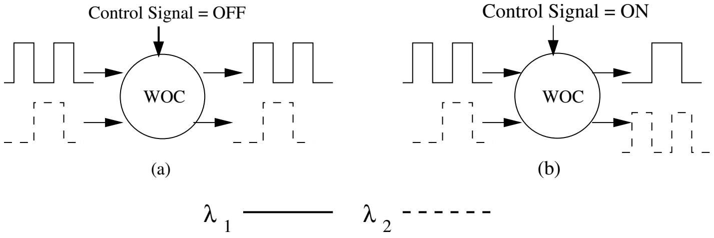


Fig. 1. The WOC device and its different configurations: (a) Bar state (b) Simultaneous switching and wavelength conversion

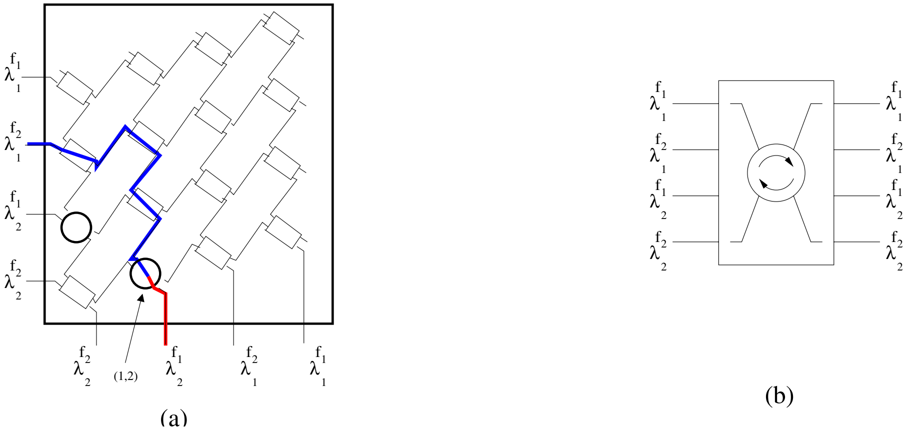


Fig. 2. (a) The $2^{\lambda}(2 \times 2)$ WDM crossbar switch, (b) Symbolic notation

3.2 WDM Crossbar Switches

Let $W^{\lambda}(F{\times}F)$ denotes an $N \times N$ WDM interconnect with F input and F output fibers, where each fiber has W wavelengths, and $N = FW$. Without lose of generality, F and W are assumed to be powers of 2. We denote by $\lambda_w^{f_j}$, a signal on wavelength λ_w in fiber j. Figure 2 shows a $2^{\lambda}(2{\times}2)$ WDM crossbar and symbolic notation. Unlike conventional crossbars where signals can be switched only in the space domain, WDM crossbars employ WOCs and thus they can switch in both space and wavelength domains. The label $(1,2)$ in the figure indicates a WOC that exchanges signal between λ_1 and λ_2.

3.3 A New WDM Interconnect Design

The basic idea in the new design is to perform pure space switching in the first and third stages while any needed wavelength conversion is performed in the middle stage. Therefore, in our design, *space* crossbars are used in the *first* and *third* stages, whereas WDM crossbars in the *middle* stage. Thus, there are r switches each of size $1^{\lambda}(n \times n)$ in the first stage $(2 \leq n \leq F-1)$; n switches each of size $(\frac{N}{F})^{\lambda}(\frac{F}{n} \times \frac{F}{n})$ in the second stage; and r switches each of size $1^{\lambda}(n \times n)$ in the third stage.

Different values of n lead to different interconnect structures, thus, there are $F-1$ different designs for a $W^{\lambda}(F{\times}F)$ interconnect. However, if we assume $n = 2^k, 1 \leq k \leq log_2F$, then there are at most log_2F designs.

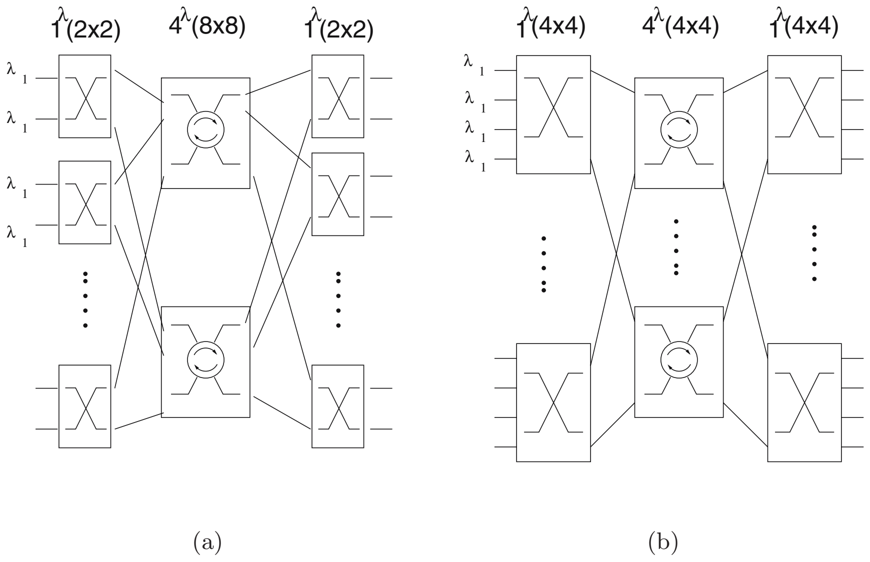


Fig. 3. Two possible designs of $4^{\lambda}(16 \times 16)$ WDM interconnect with: (a) $n = 2$, and (b) $n = 4$.

3.4 An Example: The $4^\lambda(16 \times 16)$ WDM Interconnect

Here we show a $4^\lambda(16 \times 16)$ interconnect using the proposed design approach. Since $F = 16$, hence, there are 4 ($= log_2 16$) different designs. The cost of these designs is given in Table 1. It is worth noting that the four designs have the same number of WOCs. As we show later in this paper, the number of WOCs does not depend on the value of n. Figure 3 shows two $4^\lambda(16 \times 16)$ interconnect designs for $n = 2$ and $n = 4$.

Table 1. Four different designs and their hardware cost for a $4^\lambda(16 \times 16)$ WDM interconnect

n	First	Middle	Third	# SEs	# WOCs
2	$1^\lambda(2 \times 2)$	$4^\lambda(8 \times 8)$	$1^\lambda(2 \times 2)$.	2208	96
4	$1^\lambda(4 \times 4)$	$4^\lambda(4 \times 4)$	$1^\lambda(4 \times 4)$	1440	96
8	$1^\lambda(8 \times 8)$	$4^\lambda(2 \times 2)$	$1^\lambda(8 \times 8)$	1440	96
16	$1^\lambda(32 \times 32)$	$4^\lambda(1 \times 1)$	$1^\lambda(32 \times 32)$	8512	96

4 Hardware Cost

To estimate the hardware complexity of the new design, we compute the overall number of switches and WOCs. In the following, we compute the cost of a WDM interconnect under the proposed design, and then, we drive the values of n to design an interconnect with minimum hardware cost.

Lemma 1. *the total number of SEs and WOCs in a $W^\lambda(F \times F)$ WDM crossbar is:*

$$\#SE = \frac{N}{2}(2N - W + 1) \tag{1}$$

$$\#WOC = \frac{N}{2}(W - 1) \tag{2}$$

Proof. The proof follows directly from [18].

Lemma 2. *the total number of SEs and WOCs in a $W^\lambda(F \times F)$ WDM crossbar is:*

$$\#SEs = 2Nn + \frac{N^2}{n} - \frac{N}{2}(W - 1) \tag{3}$$

$$\#WOCs = \frac{N}{2}(W - 1) \tag{4}$$

Proof. The first and third stages consist of switches of size n^2, therefore, there are $2.r.n^2$ SEs in these two stages. The number of SEs in the middle stage can *be computed by substituting N with (N/n) (the size of a WDM crossbar in the middle stage) in Equation (1). Thus, the total number of SEs is:*

$$\#SEs = 2.r.n^2 + r.\frac{N}{2n}.(\frac{2N}{n} - \frac{N}{F} + 1) \tag{5}$$

Substituting r with N/n in the above equation, we obtain:

$$\#SEs = 2Nn + \frac{N^2}{n} - \frac{N}{2}(W - 1) \tag{6}$$

The number of WOCs depends only on the size of the switches in the middle stage. It is straightforward to show that the total number of WOCs in the new design is:

$$\#WOCs = \frac{N}{2}(W - 1) \tag{7}$$

Since different values of n result in different designs with different hardware costs, it is interesting to investigate the value of n that leads to an interconnect with minimum hardware cost. It may be noted from (7) that the number of WOCs is independent on the value of n. Therefore, to optimize the cost of an interconnect under our design, it is sufficient to minimize the number of SEs.

To find the optimal value of n that minimizes the total number of SEs, we set the derivation of # SEs in Equation (6) to zero, to obtain:

$$n = \frac{1}{\sqrt{2}}.N^{\frac{1}{2}}. \tag{8}$$

By substituting the value of n above in the total number of SEs in Equation (6), we obtain:

$$\#\text{ SEs} = (2N)^{\frac{3}{2}} - \frac{N}{2}(W - 1) \tag{9}$$

5 Comparison of Designs

Recent WDM interconnect designs with *sparse* crossbar switches can potentially reduce the number of SEs and have shown to be very cost-effective compared to existing interconnect designs [38] [39]. Therefore, it is sufficient to compare our design with these two recent designs (See Table 2). The *Sparse/FWC* interconnect design in [39] requires the *minimum* number of SEs compared to existing WDM interconnects including the proposed. However, the design in [39] requires FW Full-range WCs ($|S| = |D| = W$), leading to $O(W^3)$ conversion cost, which makes the design impractical or economically infeasible when the number of wavelengths increases.

To reduce the conversion cost while controlling the number of SEs, the *Sparse/LWC* interconnect [38] make use of LWCs and sparse crossbar switches.

Sparse/LWC employs Fixed-range WCs ($|S| = |D| = 1$) at the input of the interconnect to convert all input signals to a specific wavelength. At the output of the interconnect, LWCs with conversion cost c ($|S| = 1$ and $|D| = c$, $1 \leq c \leq W$) are used in order to convert output signals to the required wavelength.

Clearly, *Sparse/LWC* interconnects are more practical as compared to the *Sparse/FWC* designs, and hence, in the following, we focus on comparing our design with the *Sparse/LWC* interconnects. We show that, for the same conversion cost, our design requires a smaller number of SEs compared to the *Sparse/LWC* design. This can be shown by computing the value of c by equating the conversion cost of both designs:

$$\frac{N}{2}(W-1) = N(c+1) - F \text{ , thus:} \tag{10}$$

$$c = \frac{W-3}{2} + \frac{1}{W} \tag{11}$$

Table 2. A Summary of # Crosspoints, and Conversion Cost of the different WDM interconnect designs. ($1 \leq c \leq W$).

Design	# Crosspoints	Conversion Cost
S/FWC [39]	$(2N)^{3/2} - N(W-1)$	NW^2
S/LWC [38]	$(2N)^{3/2} - N(c-1)$	$N(c+1) - F$
New	$(2N)^{\frac{3}{2}} - \frac{N}{2}(W-1)$	$\frac{N}{2}(W-1)$

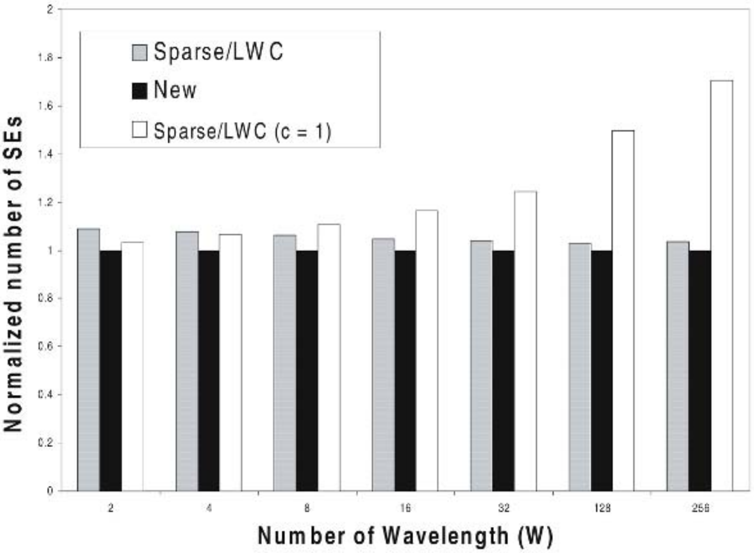


Fig. 4. The normalized number of SEs of new and Sparsedesigns for different values of W and $F = 16$

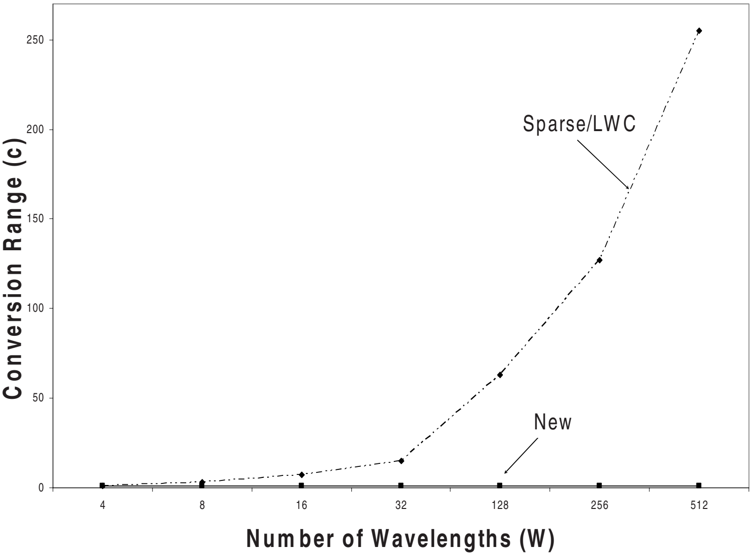


Fig. 5. The value of the conversion-range c as a function of the number of wavelengths ($F = 16$)

Using the above value of c, Figure 4 shows the total number of SEs for the *Sparse/LWC* design normalized to the total number of SEs in our design for an interconnect with $F = 16$ and for different number of wavelengths. As shown in the figure, our design has smaller number of SEs compared to the *Sparse/LWC* design. Also, it may be noted that the conversion range, c, for the LWCs in *Sparse/LWC* design increases as the number of wavelengths increases (See Figure 5) even when both designs have the same overall conversion cost. Moreover, as conversion range increases the WCs become harder to implement with current optical technologies, limiting the scalability of the *Sparse/LWC* design.

It may be noted that a *Sparse/LWC* interconnect can be designed using only *fixed-range* WCs by selecting $c = 1$ [38]. Such designs render smaller conversion cost by increasing the number of SEs (See Figure 4). As the number of wavelengths increases, this design may require more than 1.5 times the number of SEs in our design.

It should be pointed out, however, that, an accurate comparison of different designs should take into consideration the *overall* cost of an interconnect. Indeed, since the proposed design and the *Sparse/LWC* are based on two different technologies, thus, the actual costs of WOCs, WCs, and SEs, will determine which of the two designs has a smaller overall hardware cost. For example, if the cost of WOCs is higher than that of WCs, then, the *Sparse/LWC* design will probably have a smaller overall cost compared to the proposed design. Such analysis is difficult since we do not have a good estimation of the actual cost of WOCs, however, one can analyze a range of values for each of the different components (i.e. WOCs, WCs, and SEs) and identify the regions in which each design will have a smaller cost, we leave this investigation for future work.

Although, we have focused our comparison on the number of SEs and the conversion cost, it is worth noting that sparse crossbars require complex routing algorithms compared to full crossbars [38]. Our design, on the other hand, preserves the structure of full crossbars, and hence, any existing routing algorithm can be readily adapted for our design. Therefore, the new design provides *"fast and simple"* switching.

6 Conclusions

We propose a new WDM optical interconnect design that provides full-connectivity while performing conversion between *predefined* wavelengths, and hence, eliminating the need for expensive full- and wide-range wavelength converters used in most designs. This not only reduces the conversion cost, but also provides *fast and simple* switching. Moreover, the conversion range is independent of the number of wavelengths in the network which improves the scalability of interconnect. In addition, for the *same conversion cost*, the proposed design requires a smaller number of SEs compared to best known designs.

Acknowledgment

This work was supported, in part, by an NSF EPSCoR grant EPS- 0346476 and by a Nebraska Research Initiative grant. The authors would like to thank the anonymous reviewers for their suggestions for improving this paper.

References

1. A.C. Dasylva, D.Y. Montuno, and P. Kodaypak, "Nonblocking space-wavelength networks with wave-mixing frequency conversion," *J. Opt. Netw.* 1, pp. 206-216, 2002.
2. A. Chowdhury, S.C. Hagness, and L. McCaughan, "Simultaneous optical wavelength interchange with a two-dimensional second-order nonlinear photonic crystal," *Opt. Lett.,* vol. 25, No.11, June 2000, pp. 832-834.
3. A. Rasala and G. Wilfong, "Strictly non-blocking WDM cross-connects," *In Proc. of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'02),* pp. 606-615, 2000.
4. B. Mukherjee, "WDM optical communication netowrks: progress and challenges," *IEEE JSAC.*, vol. 18, no. 10 pp. 1810-1824, 2000.
5. C. Clos, "A study of non-blocking switching networks," *Bell System Tech. J.*, pp. 407-424, 1958.
6. C. Qiao and M. Yoo, "Optical burst switching (obs)- a new paradigm for an optical internet," *J. of High Speed Networks,* vol.8, no.1, pp. 69-84, 1999.
7. D.C. Opferman and N.T. Tsao-Wu, "On a class of rearrangeable switching networks, Part I: control algorithm," *Bell Syst. Tech. J.,* vol. 5, no. 50, pp. 1579-1600, 1971.
8. D. Pan, V. Anand, and H.Q. Ngo, "Cost-effective constructions for nonblocking WDM multicast switching networks," *IEEE ICC 04*, pp. 1801-1805

9. F.K. Hwang, “A survey of nonblocking multicast three-stage Clos networks,” *IEEE Com. Mag.,* pp. 34-37, 2003.
10. G.R. Hill, et al., “A transport network layer based on optical netowrk elements,” *J. Lightwave Tech.* vol 11, pp. 667-679, May/June 1993.
11. G. Wilfong, B. Mikkelsen, C. Doerr, and M. Zirngibl, “WDM cross-connect architectures with reduced complexity,” *J. of Lightwave Tech..* vol. 17, no. 10, 1999, pp. 1732-1741.
12. G. Xiao and Y.W. Leung, “Algorithms for allocating wavelength converters in all-optical networks,” *IEEE/ACM Trans. on Networking,* vol. 7, pp. 545-557, 1999.
13. H. Jonathan Chao, K-L. Deng, and Z. Jing, “PetaStar: A Petabit photonic packet switch,” *IEEE JSAC.,* vol. 21, no. 7, pp. 1096-1112, 2003.
14. H. Jonathan Chao, Z. Jing, and S.Y. Liew, “Matching algorithms for three-stage bufferless Clos network switches,” *IEEE Communication Mag.,* pp. 46-54, 2003.
15. H.Q. Ngo, D. Pan, and C. Qiao, “Nonblocking WDM switches based on arrayed waveguide grating and limited wavelegnth conversion,” *Proc. 23rd IEEE INFOCOM 04’,* 2004.
16. H.Q. Ngo, D. Pan, and Y. Yang, “Optical switching networks with minimum number of limited range wavelength converters,” *Proc. 24rd IEEE INFOCOM 2005,* 2005.
17. H.S. Hamza and J.S. Deogun, “Wavelength exchanging Cross-Connect (WEX)- a new class of photonic cross-connect architectures,” *IEEE/OSA J. Lightwave Tech.*, (*To appear*).
18. H.S. Hamza and J.S. Deogun, “WDM Optical Interconnects – A Balanced Design Approach,” *Manuscript under review.*
19. H. S. Hinton, “A nonblocking optical interconnection network using directional couplers,” *GLOBECOM 1984*, pp. 26.5.1-26.5.5, 1984.
20. J. Cheyns *et al.,*“Clos lives on in optical packet switching,” *IEEE Com. Mag.,* pp. 114-121, 2003.
21. J. Ramamirtham and J.S. Turner, “Design of wavelength converting switches for optical burst switching,” *in Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 02’)*, vol. 2, pp. 1162-1171, 2005.
22. J.Y. Hui, Switching and traffic theory for integrated broadband network, “Point-to-point multistage circuit switching,” Kluwer, pp. 53-83, 1990.
23. K.-C. Lee and V.O.K. Li, “A wavelength-convertible optical network,” *J. of Lightwave Tech.,* vol. 11, pp. 962-970, 1993.
24. K. Moei, H. Takara and M. Saruwatari, “Wavelength interchange with an optical parametric loop mirror,” *Electronics Lett.,* vol.33, no.6, pp. 520 -522, Mar 1997.
25. K. Uesaka, K. K-Y. Wong, M.E. Marhic, and L.G. Kazovsky, “Wavelength Exchange in a Highly Nonlinear Dispersion-Shifted Fiber: Theory and Experiments,” *IEEE J. of Selected Topics in Quantum Electronics,* vol. 8, no. 3, pp. 560-568, May/June 2002.
26. K. Zhu, H. Zang, and B. Mukherjee “A comprehensive study on next-generation optical grooming switches,” *IEEE JSAC.,* vol. 21, no. 7, pp. 1173-1186, 2003.
27. N. Antoniades, S.J.B. Yoo, K. Bala, G. Ellinas, and T.E. Stern, “An architecture for a wavelength-Interchanging cross-connect utilizing parametric wavelength converters,” *J. of Lightwave Tech.*, vol. 17. no. 7, July 1999.
28. N.P. Torrington-Smith, H.T. Mouftah and M.H. Rahman, “An evaluation of optical switch architectures utilizing wavelength converters,” *Electrical and Computer Eng., Canadian Conf.*, vol. 2 pp. 1008 -1013, 2000.

29. R.A. Barry and P.A. Humblet, "Models of blocking probability in all-optical networks with an without wavelength changers, " *IEEE J. Selected Areas in Communications,* vol. 14, pp. 858-867, 1996.
30. R. Kannan, "The KR-Benes network: a control-optimal rearrangeable permutation network," *IEEE Tran. on Computers,* vol. 54, no. 5, pp. 534-544, 2005.
31. S. Subramaniam, M. Azizoglu, and A.K. Somani, "On optimal converter placement in wavelength-routed networks, " *IEEE/ACM Tran. on Networking,* vol. 7, pp. 754-766, 1999.
32. T.E. Stern and K. Bala. Multiwavelength optical networks: a layered approach. Addison Wesley, 1999.
33. T.T. Lee and S.Y. Liew, "Parallel routing algorithms in Benes-Close networks," *IEEE Tran. on Communications,* vol. 50, no. 11, pp. 1841-1847, November 2002.
34. V.E. Benes, "On rearrangeable three-stage connecting networks," *Bell Syst. Tech. J.*, vol. XLI, no. 5, Sept. 1962.
35. W. J. Dally and B. Towles. Principles and practices of interconnection networks. Morgan Kaufmann Publishers, 2004.
36. X. Qin, and Y. Yang, "Nonblocking WDM switching networks with full and limted wavelegnth conversion," *IEEE Transactions on Communications*, vol. 50, no. 12, pp. 2032-2041, 2002.
37. Y. Yang, J. Wang, and C. Qiao, "Nonblocking WDM multicast switching networks," *IEEE Tran. on Parallerl and distributed systems,* vol. 11, no. 12, Dec. 2000, pp.1274-1287.
38. Y. Yang and J. Wang, "Desiging WDM optical interconnects with full connectivity by using limited wavelength conversion," *IEEE Transactions on Computers,* vol. 53, no. 12, pp. 1547-1556, 2004.
39. Y. Yang and J. Wang, "Cost-effective designs of WDM optical interconnects," *IEEE Transactions on Parallel and Distributed Sys.,* vol. 16, no. 1., pp. 51-66, 2005.

Designing Fast and Bandwidth Efficient Protection Scheme for WDM Optical Networks

Yu Lin, Haitham S. Hamza, and Jitender S. Deogun

Department of Computer Science & Engineering,
University of Nebraska-Lincoln Lincoln, NE 68588-0115, USA
{ylin, hhamza, deogun}@cse.unl.edu

Abstract. In this paper, we introduce the *Pre-cross-connected Segment First* — PXSFirst protection scheme; a new path-based protection schemes for WDM optical networks. The goal of the PXSFirst scheme is to achieve fast recovery, while maximizing bandwidth sharing to improve bandwidth utilization. Similar to pre-cross-connected trails (PXTs) protection scheme, PXSFirst ensures that all backup paths are pre-cross-connected, and hence eliminates the switching configuration delay along backup paths. However, unlike the PXT scheme, where backup paths can only share trails, PXSFirst breaks backup paths into smaller segments by existing end nodes, and hence, increases the possibility of bandwidth sharing. Extensive simulation results for different network topologies and under different traffic patterns show that the proposed scheme has better blocking performance and less bandwidth utilization (an average of 11.0% reduction) compared to existing PXT protection schemes.

1 Introduction

The design of a survivable Wavelength Division Multiplexing WDM network that provides fast recovery with minimum network resources is an important problem. The significance of the problem increases considerably as (WDM) technology matures and more and more wavelengths per fiber become available, making even the failure of a single fiber catastrophic. Under single failure model, a wavelength-routed optical network may fail due to the failure of a single component (e.g. link or node) of the network. Such a component failure will result in a failure of all the connections those utilize that component (e.g. link).

Traditionally, ring protection schemes (e.g. SONET) are used to ensure fast recovery for any single link or node failure, due to the fact that traffic can be rerouted around a *single* node upon any single link or node failure. However, ring protection schemes do not utilize bandwidth efficiently. As a result, techniques that aim at effectively utilizing network bandwidth has been investigated. Most notable is the shared mesh protection scheme that enables more efficient utilization of network resources [1], [6], [11]. In mesh *shared protection* methods, backup resources are *pre-allocated* during the connection setup, where two or more backup lightpaths may share one backup path, given that the corresponding primary lightpaths are link and node disjoint. The sharing of backup paths, however, can lead to slow recovery. This is because switching nodes along the

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 391–402, 2006.

shared backup paths may need to be reconfigured conditionally based on which particular primary path needs to be recovered.

An effective protection scheme, therefore, should attain the bandwidth efficiency of a mesh protection while providing a ring-like recovery speed [11], as well as to ensure transmission integrity. One of the recent approaches to accomplish this is the *pre-cross-connect trail* (PXT) [11]. The idea of PXT is to provision the network such that all protection paths are pre-cross-connected for prompt use upon failure [11]. However, under dynamic traffic, some nodes that are pre-cross-connected under the PXT scheme may become end nodes of a new request and thus cannot be pre-cross-connected when provisioning future requests. As a result, by arranging protection edges into pre-cross-connected trails, which is equivalent to keep track of the complete protection part of the network, some pre-cross-connected segments may fail to be reused, leading to less efficient utilization of bandwidth in the network.

Therefore, in order to improve bandwidth utilization, we need to reuse any pre-cross-connected segments along the backup paths. We thus believe that to ensure fast restoration while providing efficient bandwidth utilization, neither ring nor trails are sufficient, but rather we need to provision the network based on the *pre-cross-connected segments*, where a *segment* is one or more consecutive links that connects a pair of nodes in the network. In this paper, we therefore propose a new protection scheme that can ensure fast recovery while providing efficient bandwidth utilization. The new scheme is called Pre-cross-connected First (PXSFirst) scheme. PXSFirst forces the primary path to take the route which is segmented by existing end nodes.

The reminder of the paper is organized as following. Section 2 summarizes related work in fast protection schemes. The proposed algorithm is presented in Section 3. Simulation results are discussed in Section 4; Conclusions are presented in Section 5.

2 Related Work

Two main techniques have been proposed to achieve a mesh-like bandwidth efficiency, while providing a ring-like recovery speed: the $p-cycles$ [12], and the *pre-cross-connect trail* (PXT)[11].

The basic idea of $p-$cycles is to route the primary traffic using an arbitrary mesh routing algorithm, but to constrain the protection routes to lie on certain predetermined rings. Over the last few years, the basic theory of $p-$cycles has evolved and several extensions have been proposed, e.g. [2], [3], [13], [14]. However, $p-$cycles are more suitable for *static* traffic and for link-based protection schemes, also, to achieve high bandwidth utilization more complex management algorithms are needed. In [4], the concept of $p-$cycles was generalized to path segment protection, which includes the pure path-based protection as a special case. Although a technique for provisioning dynamic demands was discussed in [4], the overhead and complexity of the mechanism make it impractical for large networks.

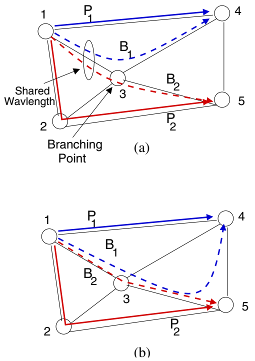


Fig. 1. Basic concept of branch points in PXT approach

In recent work [11], it has been observed that achieving fast recovery is not confined to networks with a ring-like topology. Instead, the key for fast recovery resides in the ability to pre-connect protection paths, and thus, no reconfiguration of switching nodes along backup paths is needed. Accordingly, the concept of *pre-cross-connect trail* (PXT) was proposed. The basic concept of PXT is to provision backup paths with no *branch* points. We illustrated this concept by the following example.

Consider the network in Figure 1-a. Two requests need to be provisioned. The first request is from node 1 to node 4; $(1, 4)$, while the second request is from node 1 to node 5; $(1, 5)$. Using a conventional path-based protection scheme with backup sharing, a possible provisioning of primary and backup paths for the two requests is given in Figure 1-a, where P_i and B_i represent, respectively, the primary and backup paths allocated to request i. As shown in figure, the two backup paths share a wavelength on link $(1 - 3)$. At node 3, each backup path requires connection to a different output. Therefore, node 3 is considered to be a *branch point* in this network as the configurations of node 3 upon the failure of P_1 or P_2 are different. To reduce recovery time, PXT necessitates the elimination of such branch points, so that every node in the network can be pre-connected independent of the failure of a specific primary path. For example, Figure 1-b gives another possible path provisioning for the two requests $(1, 4)$ and $(1, 5)$. As shown in the figure, B_1 does not branch at node 3 as in Figure 1-a, and hence, node 3 is no longer a branch point.

Several recent work has investigated the concept of segment-based protection. In [15], a segment based protection scheme was proposed based on dynamic programming, however in this scheme, a link may be protected by two backup segments and therefore the overlapped protection wastes bandwidth. Moreover, the end nodes of a segment, exclusive of the source and destination node of the request, must be protected by a backup segment, and hence, more bandwidth is needed.

As discussed in [15], the heuristic algorithm proposed in [16] to determine segments cannot efficiently deal with some real trap scenarios and in addition does not consider backup bandwidth sharing until the paths are found. [15] also suggests the scheme in [17] requires the node immediately upstream from the link/node failure to restore traffic along an alternate outgoing link, which limits its flexibility. [18] divides the primary path into segments of equal length and is therefore not flexible. Furthermore, in order to protect the last link and the end node incident to the last link, the last link of each segment is under the protection of two backup segments, which implies the scheme is not bandwidth efficient either.

3 The Proposed Algorithm

Before we present the formal algorithm of the proposed protection scheme, we first illustrate the underlying concepts of the proposed protection scheme.

3.1 The Basic Concept

We now introduce the main insight of this paper. In all path-based protection scheme, source nodes and destination nodes are assumed to be invulnerable, since the protection scheme needs the end nodes to react to the failure and perform a real time switching to reroute the traffic to the backup path. Therefore a protection segment ends in any two end nodes along the backup paths may be reused. Given the existing pre-cross-connected segments, we want to force the primary path to take the route which is divided by the end nodes in a way such that the resulting segments along the primary path are already under protection and are link and node disjoint with other segments under the the protection of the same backup segment. We denote the set of primary segments protected by the same backup segments as a conflict set.

Now, we illustrate the difference between the proposed protection scheme and the conventional shared pre-cross-connected protection scheme with an example. In Figure 2, solid lines represent physical links, dashed lines represent the primary light paths and dotted lines represent the backup light paths of the requests upon link or node failure under the proposed PXSFirst protection scheme. Figure 3 represents the same network as Figure 2 represents except that the conventional shared pre-cross-connected protection scheme is applied. Each link consists of a single wavelength channel and each link has a unit capacity.

Upon the first request (A, C), path (A, B, C) and (A, G, C) are allocated as primary path and backup path for this request respectively.

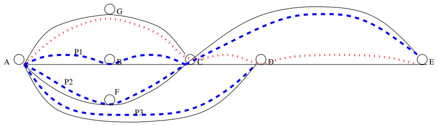


Fig. 2. Basic concept of PXSFirst

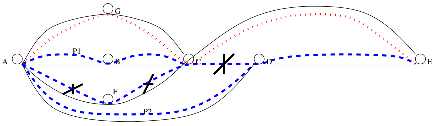


Fig. 3. Basic concept of conventional shared pre-cross-connected protection scheme

Upon the second request (A, E), the conventional shared pre-cross-connected protection scheme (Fig. 3) applies the shortest path and allocates (A, D, E) as the primary path. It further allocates (A, G, C, E) as the backup path. We will see shortly that this allocation blocks the third request.

Under the proposed PXSFirst (Fig. 2), however, we want to reuse the existing pre-cross-connected backup segments as much as possible and therefore take the path (A, F, C, E) as the primary path, since the segment (A, F, C) is already under protection. Path (A, G, C, D, E) is further allocated as the backup path.

Now suppose the third request (A, D) arrives. Under the conventional shared pre-cross-connected protection scheme, (A, F, C, D) is the only available path from A to D, and there is no way to allocate a backup path for this primary path. The request is thus blocked due to the lack of backup path.

However under the proposed PXSFirst, path (A, D) can be allocated as the primary path and path (A, G, C, D) is reused as backup path. PXSFirst thus successfully escapes the trap by utilizing the segment already under protection. In this example, if segment (A, F, C) is not allocated to the second request, which is the case under the conventional shared pre-cross-connected protection scheme, the segment can be isolated and can thus be wasted, as illustrated by Fig. 3.

3.2 The PXSFirst Protection Algorithm

In the proposed algorithm, we assume the following constraints while provisioning resources for each request:

C1: the current primary path is link disjoint with the existing primary paths.
C2: the current primary path is link disjoint with the existing protection paths.
C3: all links and nodes along the current primary path are protected except the end nodes.
C4: existing protection edges are reused as much as possible without introducing branch points.

Before we present the details of the algorithm, we first provide a few definitions and notations. Let V_m denote the *conflict set* for a pre-cross-connected segment m. A conflict set V_m contains all primary segments whose backup path traverse the pre-cross-connected segment m. Therefore two primary segments whose backup paths traverse m must be node and link disjoint. It is also obvious that $V_m \neq \emptyset$ implies the segment m belongs to a backup path. Table 3.2 summarizes the notations and variables used in this paper.

Table 1. Summary of notations and variables used in this paper

Notations	Definition
m	A segment, which can be an edge; a consecutive set of edges; or an end-to-end path.
V_m	The conflict set V_m of segment m.
A_r	The set of segments that are link and node disjoint with the current primary path, exclusive of the end nodes of the current request r; and link disjoint with the existing primary paths.
M	The set of pre-cross-connected segments resulting from dividing the backup paths by end nodes.
p_m	Given $m \in M$, the shortest path between the end nodes of m, link and node disjoint with any segments in V_m
M'	$\bigcup_{m \in M} p_m$

Upon a request arrival, PXSFirst factors out, with all combinations of two end nodes, any new segments along the backup paths . The resulting segments, along with the primary paths these segments protect, are appended to the global set M, such that all segments in M are reusable by the future requests.

To provision a primary path for the request, PXSFirst finds a shortest path between the two end nodes of every pre-cross-connected segment m. The shortest path must be link and node disjoint with the primary segments in V_m. The set of shortest paths for all m in M is denoted as M'. PXSFirst then constructs an auxiliary graph P with the same set of nodes as in G, which is the original graph representing the network. The cost of link e in G is denoted as $C_G(e)$. Graph P's link cost function, denoted as $C_P^E(e)$, associates link e in P with the set of segments E in G that share the same end nodes as e in P. Since we want the primary path to take the route that has already been protected to the maximum extent possible, we multiply $C_P^E(e)$ by ϵ, where $\epsilon < 1.0$, if any segment in E, to which e corresponds to, is in M'. Otherwise, we want the primary path to take the route in G, which is link disjoint with the existing primary paths. The link cost function can be formalized as follows:

$$C_P^E(e) := \begin{cases} \epsilon, & \text{if } \exists m \in E, m \in M'; \\ C_G(e), & \text{Otherwise.} \end{cases} \quad (1)$$

PXSFirst then applies the shortest path algorithm with the new relax step on P. The new relaxation step, described in Fig. 4, is to ensure the resulting path in P is indeed a path in G, that is the segments in G corresponding to the links along the resulting path in P after expansion do not overlap in G.

During the second phase of the PXSFirst scheme, we allocate a backup path, that confirms to the following constraints: a) the current primary path and backup path for request r are node disjoint except for any end nodes, b) the

```
RELAX($u$, $v$)

1. Expand all links on the partially available primary path/backup path to the corresponding segments in $G$, the set of nodes along the segments form $S_1$;
2. Expand the link from $u$ to $v$ to the corresponding segment in $G$, the set of nodes along the segments form $S_2$ ;
3. If $S_1$ and $S_2$ intersects on a node other than $u$, continue;
4. If $c^v > c_u^v + c^u$
5. Set node $u$ as node $v$'s previous hop;
6. Append $S_2$ to $S_1$.
```

Fig. 4. RELAX Step in a Standard Shortest Path Algorithm

backup path for request r and the set of existing primary paths are link disjoint, c) backup sharing does not introduce branch points. We construct another auxiliary graph H that captures these constraints and the link cost function of H is denoted as $C_H^E(e)$, where e in H is associated with the set of segments in G, which share the same end nodes as e. The afore-going constraints are captured by the second case in the cost function below. The first case of the link cost function ensures the maximum backup paths sharing. Otherwise the backup path is free to use any idle link in G, as implied by the third case of cost function.

$$C_H^E(e) := \begin{cases} 0, & \text{if } \exists m \in E, m \in M \wedge p(r) \notin V_m; \\ +\infty, & \text{else if } \forall m \in M, m \in \bar{A}_r; \\ C_G(e), & \text{Otherwise.} \end{cases} \tag{2}$$

Overall, the proposed PXSFirst (see Fig. 5) applies shortest path algorithm with the new relaxation step on the auxiliary graphs P and H during the primary path and backup path allocation phase. For any node v in P/H, we associate a cost variable c^v, which denotes the cost of the primary path/backup path from the starting source node to v and a cost variable c_u^v, which denotes the cost from u to v. As the shortest-path algorithm progresses on P/H, we maintain a set $S1$ containing the nodes on the segments in G, that correspond to the links on the partially available primary path/backup path, with the new relaxation step.

In PXT algorithm, the resulting path in H expands to a trail in G, as the name of the algorithm alludes. PXT algorithm then stores the trail in L_1 and removes the cycles on the trail to output a backup path. We argue it is not efficient from the storage and algorithm simplicity point view and claim it is more meaningful to embed the cycle control within the standard shortest-path algorithm with the new relaxation step.

4 Numerical Evaluation

In this section, we experimentally evaluate the performance of the proposed algorithm on the Italian network (not shown here), the Pacific Bell network, and

```
PXSFirst
Input: Graph $G$, request $r$, set $M$ as defined in the table of notations and variables.
Output: p(r) and b(r) satisfying the constraints C1- C4.

1. Update set $M$ by factoring out any new segments that ends in any combination
   of end nodes.
2. Construct graph $P$ with link cost function $C_P^E(e)$;
3. Apply shortest path algorithm on $P$ with the new relaxation step and obtain $p(r)$;
4. Update $G$ by marking the links used by $p(r)$;
5. Construct graph $H$ with link cost function $C_H^E(e)$;
6. Apply shortest path algorithm on $H$ with the new relaxation step and obtain
   $b(r)$;
7. Update $G$ by marking the links along $b(r)$, which are not in use before .
```

Fig. 5. The proposed PXSFirst algorithm

a 12-node random network. Moreover, to validate the findings, we also used a 12-node ring, bipartite, grid, and clique topologies. For each topology, we simulated three different request models [11]: *Uniform Requests*, a *Nearest-Neighbor Requests*, and *Unbalanced Requests*. For experimental purpose we set ϵ in Equation (1) to zero. We obtained the results by running the algorithms once on each distinct sequence of traffic demand arrivals.

4.1 Assumptions

In this study, we consider an incremental request model, where requests are fed one at a time to the algorithm; Full-range Wavelength converters, where an input wavelength can be converted to any output wavelength; a fixed set of routes between every pair in the network are pre-computed and are not affected by the dynamics of network; and a single failure model [7].

Three different request models are considered [11]:

1. *Uniform Requests.* Every pair of nodes appears five times in the request list.
2. *Nearest-Neighbor Requests.* Every pair of neighboring nodes appears 5 times.
3. *Unbalanced Requests.* We choose three arbitrary nodes as *Critical* (C) nodes and the rest of the nodes as *Normal* (N) nodes. Critical nodes may represent nodes that groom high-priority traffic, for example. The combination of a C node with an C node, a C node with an N node, and an N node with an N node appear, respectively, 12, 8, and 5 times in the request list.

4.2 Evaluation Metrics

We use five metrics to evaluate the performance of the proposed algorithm:

1. *Blocking Probability.* The blocking probability, b_p, for a set of requests, R:

$$b_p = \frac{\#\text{of blocked requests}}{|R|} \tag{3}$$

A request is blocked if there is no available primary path or available backup path that can be allocated to this request.

2. *Bandwidth Utilization.* The bandwidth utilization, b_u, for a given protection scheme is defined as:

$$b_u = \frac{\sum\limits_{\forall e \in E} \text{\#of used wavelengths on}e}{|E|.W} \tag{4}$$

b_u measures the number of W used to provision all requests.

3. *Mean Primary Path length.* The mean of primary paths length, $|\overline{p(r)}|$, for a given protection scheme is defined as:

$$|\overline{p(r)}| = \frac{\sum\limits_{\forall r \in R} |p(r)|}{|R|}, \tag{5}$$

where R is set of all requests

4. *Mean Backup Path length.* The mean of backup paths length, $|\overline{b(r)}|$, for a given protection scheme is defined as:

$$|\overline{b(r)}| = \frac{\sum\limits_{\forall r \in R} |b(r)|}{|R|}. \tag{6}$$

Table 2. Blocking probability (b_p) and Bandwidth utilization (b_u) of different protection schemes ($W = 20$). Only instances in which the PXT and PXSFirst have $b_p = 0$ are considered.

Simulation Parameters			**Blocking Probability**				**Bandwidth Utilization**		
Network	Traffic	# Req	1+1	PXT	Shared Path	PXSFirst	PXT	Shared Path	PXSFirst
bipartite	uniform	660	51.8	27.4	9.5	6.0	—	—	—
	nearestNB	360	18.1	0.0	0.0	0.0	30.0	26.3	17.5
	unblanced	864	63.9	41.2	20.7	17.0	—	—	—
clique	uniform	660	5.9	0.0	0.0	0.0	30.0	25.9	17.3
	nearestNB	660	7.1	0.0	0.0	0.0	30.0	25.8	17.2
	unblanced	864	15.9	0.0	0.0	0.0	37.7	33.5	22.3
ring	uniform	660	96.2	91.2	75.6	82.0	—	—	—
	nearestNB	120	69.2	0.0	0.0	0.0	30.0	30.0	19.6
	unbalanced	864	97.5	93.8	88.8	83.7	—	—	—
grid	uniform	660	88.5	76.1	55.3	61.8	—	—	—
	nearestNB	165	21.2	0.0	0.0	0.0	30.0	30.0	18.8
	unbalanced	864	90.4	80.0	65.2	68.1	—	—	—
random	uniform	660	81.5	70.5	45.9	50.2	—	—	—
	nearestNB	200	5.0	0.0	0.0	0.0	30.0	30.0	27.5
	unbalanced	864	86.5	78.9	58.9	60.1	—	—	—
Pacific	uniform	1050	92.8	84.7	67.3	74.0	—	—	—
	nearestNB	210	21.0	0.0	0.0	0.0	30.0	30.1	19.0
	unbalanced	1308	92.4	87.2	71.8	76.5	—	—	—

4.3 Results

We conducted experiments on networks with 10 and 20 wavelengths per fiber. The results follow the same trends, thus we only show results for the 20 wavelengths case. For comparison, we also implemented the traditional 1+1, the original PXT and the Shared Path protection scheme. In the following, we summarize our main results.

1. *Blocking Probability* (p_b). Table 2 gives the (p_b) of the four protection schemes for the different topologies and under the three request models. As shown in the tables, the new algorithm has the best performance compared to 1+1 and PXT in all cases and Shared Path protection schemes for bipartite and ring topology under certain traffic patterns. Comparing to the PXT protection scheme, PXSFirst reduces the blocking probability by an average of 8.75%.
2. *Bandwidth Utilization* (b_u). We compared b_u for cases in which both the PXT, Shared Path and PXSFirst protection scheme have zero blocking (i.e. all requests were successfully provisioned). Table 2 shows the values of b_u for different topologies. As shown in the table, PXSFirst provides better bandwidth utilization compared to PXT for all the topologies. Under the selective request models, the improvement reaches an average of 11.0%

Table 3. Mean Primary Path Length ($|\overline{p(r)}|$) and Mean Backup Path Length ($|\overline{b(r)}|$) of different protection schemes for ($W = 20$). Only PXT and PXSFirst protection schemes are considered.

Simulation Parameters			$|\overline{p(r)}|$		$|\overline{b(r)}|$	
Network	Traffic	# Req	PXT	PXSFirst	PXT	PXSFirst
bipartite	uniform	660	1.62	1.51	1.53	3.38
	nearestNB	360	1.0	1.0	1.0	3.36
	unblanced	864	1.58	1.55	1.52	3.72
clique	uniform	660	1.0	1.0	1.0	3.46
	nearestNB	660	1.0	1.0	1.0	4.20
	unblanced	864	1.0	1.0	1.0	3.44
ring	uniform	660	2.76	2.80	6.34	4.78
	nearestNB	120	1.0	1.0	1.0	2.25
	unbalanced	864	2.66	2.44	6.21	4.70
grid	uniform	660	1.93	2.06	2.93	3.93
	nearestNB	165	1.0	1.0	1.0	2.82
	unbalanced	864	1.98	2.03	2.17	4.85
random	uniform	660	1.94	2.05	3.63	2.58
	nearestNB	200	1.0	1.0	1.0	3.0
	unbalanced	864	1.96	2.05	2.32	3.73
Pacific	uniform	1050	1.97	2.12	2.70	3.27
	nearestNB	210	1.0	2.0	1.0	1.9
	unbalanced	1308	1.90	3.02	2.53	3.48

3. *Mean Primary Path Length ($|\overline{p(r)}|$).* In order to assess the tradeoff for lower blocking probability and better bandwidth utilization, we also compared the mean primary path length for the PXT and PXSFirst protection scheme. As Table 3 shows, the mean primary path length using PXSFirst increases by an average of 0.54%, compared to that of the PXT scheme.
4. *Mean Backup Path Length ($|\overline{b(r)}|$).* While evaluating the mean backup path length, we first realize that in some topologies such as in ring and random networks, the mean backup path length under certain request models using PXSFirst is shorter than the one in PXT, whereas in other topologies or in the preceding topologies but under some request models the mean backup path length is shorter, as shown in Table 3. However the overall mean backup path length using PXSFirst increases by an average of 54.8%.

5 Conclusions

In this paper, we address the problem of developing a fast and bandwidth efficient path protection scheme for WDM optical networks. We introduce the concept of PXSFirst — Pre-cross-connected Segment First that further improves the existing PXT scheme by forcing the primary path to take the route which has been protected by pre-cross-connected segments to the maximum extent possible. Extensive simulation results of PXSFirst on several network topologies and under three different traffic models confirm an improvement in both blocking performance (an average of 8.75%) and bandwidth utilization (an average of 11.0%) compared to PXT protection scheme by a slight increase in primary path length (an average of 0.54%) and a large increase in backup path length (an average of 54.8%).

In the future, we plan to evaluate the algorithm's primary path and backup path length further. We plan to investigate the primary path length and backup path length under unlimited resources.

References

1. C. Ou, J. Zhang, H. Zang, L.H. Sahasrabuddhe, and B. Mukherjee, "New and improved approaches for shared-path protection in WDM mesh networks," *J. Lightwave Technology,* vol. 22, no. 5, pp. 1223 - 1232, 2004.
2. D. A. Schupke, C. G. Gruber, and A. Autenrieth "Optimal configuration of $p-$ cycles in WDM networks," *Proc. of IEEE ICC '02,* pp. 2761-2765, 2002.
3. D. Stamatelakis, "Theory and algorithms for preconfiguration of sparse capacity in mesh restorable networks," *M.Sc. Thesis,* University of Alberta, Canda, 1997.
4. G. Shen and W.D. Grover, "Extending the $p-$ cycle concept to path segment protection for span and node failure recovery," *J. of Selected Areas in Communications,* vol. 21, no. 8, 2003.
5. G. Mohan, C. Siva Ram Murthy, and A. K. Somani, "Efficient algorithms for routing dependable connections in WDM optical networks," *IEEE/ACM Transactions on Networking*, vol. 9, no.5, pp. 553- 566, October 2001.

6. S.-l. Kim and S.S. Lumetta, "Capacity-efficient protection with fast recovery in optically transparent mesh networks," *Proc. 1st International Conference on Broadband Networks (BroadNets 2004),* pp. 290 - 299, 2004.
7. S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM Mesh Networks," *Journal of Lightwave Techonology*, vol. 21, no. 4, 2003, pp. 870-882.
8. S. Ramamurthy and B. Mugherjee, "Survivable WDM mesh networks- Part I: Protection," *Proc. IEEE INFOCOM*, Mar. 1999, pp. 744-751.
9. S. Ramamurthy and B. Mugherjee, "Survivable WDM mesh networks- Part II: Restoration," *Proc. IEEE Integrated Circuits Conf.*, June 1999, pp. 2023-2030.
10. R. Sabella, E. Iannone, M. Listanti, M. Berdusco, and S. Binetti, "Impact of transmission performance on path routing in all-optical transport networks", *IEEE J. Select. Areas Commun.,* vol.6, pp. 1617-1622, Dec. 1988.
11. T.Y. Chow, F. Chudak, and A.M. Ffrench, "Fast optical layer mesh protection using pre-cross-connected trails," *IEEE/ACM Transactions on Networking,* vol. 12, no. 3, pp. 539-548, 2004.
12. W.D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," *Proc. IEEE Int. Conf. Communications,* pp. 537-543, 1998.
13. W. D. Grover and J. E. Doucette, "Advances inoptical network dsign with $p-$ cycles: joint opitimization and pre-selection of candidate $p-$ cycles," *Proc. IEEE-LEOS Topical Meeting,* WA2-49-WA2-50, 2002.
14. W. Grover and D. Stamatelakis, "Bridging the ring-mesh dichotomy with $p-$ cycles," *Proc. IEEE/VDE DRCN 2000,* pp. 92-104, 2000.
15. D. Xu, Y. Xiong, C. Qiao, "A New PROMISE Algorithm in Networks with Shared Risk Link Group," IEEE Globecom, San Francisco, CA, Dec. 2003
16. C.V. Saradhi and C.S.R. Murthy, "Dynamic establishment of segmented protection paths in single and multi-fiber WDM mesh networks," *OPTICOMM'02,* pp. 211-22.
17. M. Kodialam and T. V. Lakshman, "Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information," *INFOCOM'01,* 2001, pp. 376-385
18. P-H Ho and H. Mouftah, "A framework for service-guaranteed shared protection in WDM mesh networks," *IEEE Comm. Mag.,* vol. 40, no. 2, 2002, pp. 97-103

Increasing Fairness and Efficiency Using the MadMac Protocol in Ad Hoc Networks

Tahiry Razafindralambo* and Isabelle Guérin-Lassous

CITI Lab.- Project INRIA ARES,
Bât L. De Vinci - 21 av. J. Capelle - 69621 Villeurbanne - France
{tahiry.razafindralambo, isabelle.guerin-lassous}@insa-lyon.fr

Abstract. The IEEE 802.11 MAC layer is known for its unfairness behavior in *ad hoc* networks. Introducing fairness in the 802.11 MAC protocol may lead to a global throughput decrease. It is still a real challenge to design a fair MAC protocol for ad hoc networks that is distributed, topology independent, that relies on no explicit information exchanges and that is efficient, *i.e.* that achieves a good aggregate throughput. The MadMac protocol deals with fairness and throughput by maximizing aggregate throughput when unfairness is solved. Fairness provided by MadMac is only based on information provided by the 802.11 MAC layer and adds a non-probabilistic modification in 802.11. MadMac has been tested in many configurations that are known to be unfair. In these configurations, MadMac provides a good aggregate throughput while solving the fairness issues.

1 Introduction

Ad hoc networks have become more and more popular and many research problems, such as routing, quality of service, security, etc., are now addressed. Most of the current ad hoc networks are based on the IEEE 802.11 standard [8] owing to the fact that this is the most widespread technology in the field of wireless local networks and it provides a distributed medium access with the DCF mode. Recently, different studies have shown some performance issues with the DCF mode, used in ad hoc network. These studies show that the origin of the performance problems comes from the MAC layer of this mode. These performance problems often lead to unfair situations and global performance loss [5].

Several solutions have been proposed to improve 802.11 performance in wireless ad hoc networks by reducing unfairness issues or by improving global throughput. Recently, several approaches try to increase both throughput and fairness by modifying the 802.11 MAC layer. Most of these solutions are based on rate and topology information exchanged between the nodes. The proposed protocols, not based on this kind of information, either reduce the fairness issues to the detriment of the aggregate throughput or increase the overall throughput without solving the fairness issues. In [14], the authors investigate the trade-off between aggregate throughput and fairness. They propose a model to compute

* Financed by France Telecom R&D under CRE-46128746.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 403–414, 2006.

the maximum aggregate throughput under various fairness schemes, but their algorithm is based on information propagation. Therefore, it is still a real challenge to design a fair MAC protocol for ad hoc networks that is distributed, topology independent, that relies on no explicit information exchanges and that is efficient, *i.e.* that achieves a good aggregate throughput.

In this paper we propose a solution to this challenge by designing a new protocol, called *MadMac*, that increases fairness in 802.11-based ad hoc network while maintaining a good aggregate throughput in the network. One of the main advantages of MadMac is that it is easy to implement because it is only based on information provided by the 802.11 MAC layer.

In Section 2, we present a state-of-the-art on the protocols solving unfairness issues. The protocol MadMac is described in Section 3 and evaluated in several configurations that present fairness or performance issues in Section 4. We show that our protocol achieves very good performances in all these topologies and solve many problems, like for instance the performance anomaly of 802.11 [2]. Lastly, we conclude our paper with the outline of our future works.

2 Related Work

Fairness issues in ad hoc networks have been deeply studied for a couple of years. Several mechanisms and protocols have been proposed to solve the fairness issues. There exist two main approaches in the literature. One approach is based on information exchanges between stations and/or a knowledge of the topology as in [17], [11], [15], [14], [16] and [9]. The other approach is topology independent and does not required any information exchanges as in [4], [10], [1], and [7].

The authors of [15] describe a mechanism for translating a given fairness model into its corresponding collision resolution backoff algorithm that probabilistically achieves the fairness objective but requires an efficient collision avoidance scheme (as RTS/CTS) to be efficient. Results show that on ring and clique topologies the proposed protocol achieves better fairness and is more efficient than 802.11. In [14], the authors propose a packet scheduling scheme to achieve a fair and maximum allocation channel bandwidth. The algorithm proposed by the authors computes a scheduling based on a backoff modification. Their algorithm requires a knowledge of the topology and an exchange of flow information between nodes. In [16], a $p_{i,j} - persistent$ protocol where each station computes an access probability on the link between i and j is proposed. The backoff window size is computed according to information about the contention window size received from active neighbors. The authors of [11] try to enforce the max-min fairness by using an algorithm that computes the fair share. This algorithm requires the knowledge of the two-hop neighbors for each node to be efficient. In [17], the authors propose a backoff algorithm to improve both throughput and fairness. This algorithm requires the estimate of the number of active stations and a mechanism to avoid hidden terminal problem and is designed only for single hop networks. The EHATDMA protocol [9] is based on information exchanges initiated by the sender and/or the receiver before the data transmission to avoid the hidden terminal problem and leads to a better fairness than the protocol proposed in [16].

To cope with the lack of information on topology or from others nodes, some protocols base their decision on the data packets sent in the network only or introduce a probabilistic behavior in the nodes. In [1], each station adjusts its contention window size depending on its share of the medium with its neighbor nodes. This share is computed according to the number of sent packets by the station and the number of received packets from its neighbors. Results given in this paper and in [19] show that the algorithm proposed is better than 802.11 from the fairness point of view, but not from the aggregate throughput point of view. The problem with this protocol is that the share of the radio medium for a station only considers the neighbor nodes and not the nodes within the carrier sensing range. In [7] a distributed fair MAC protocol (FMAC) solves this carrier sensing problem. The main principle of FMAC is that the contention window size is tuned to reflect the number of successful transmissions during a time interval. Results given in this paper show that this protocol improves fairness but clearly reduces the network throughput. The authors of the PNAV protocol [4] introduce a fixed waiting time between two successive transmissions depending on a probability. This probability depends on past events in the network. Results on PNAV shows that PNAV improves fairness on some topologies compared to 802.11, but PNAV global throughput is always smaller than the 802.11 aggregate throughput. In [10], the authors propose a contention windows modification based on idle slots perceived by each node. This protocol is efficient in single hop networks.

Our aim is to find the best trade-off between fairness and global throughput. As far as we know, only one paper deals with the trade-off between these two notions, but the proposed algorithm requires a knowledge of the topology and an exchange of flow information between nodes [14]. We think that this approach is not the most efficient since information exchanges may reduce the global throughput of the network. For example, a mechanism like RTS/CTS, that can be considered as an information exchange between nodes, decreases the global throughput of the network. We will show for instance that, with our proposed protocol, the RTS/CTS mechanism used to solve hidden terminal problem can be replaced by an appropriate fairness scheme. However, it appears from the literature that designing a MAC protocol, fair and efficient in terms of global throughput, that does not require any knowledge of the topology or specific information from other nodes than those provided by the MAC 802.11 protocol and the data traffic in the network is still a real challenge.

Most of the algorithms proposed to improve capacity and fairness depend on a random process. This probabilistic feature is effective either on the triggering of the modification or/and on the modification process or/and on the sending of packets. For instance, in the algorithm of [15], the triggering of the modification is random, the choice of the backoff is random and the sending of a packet is random since the protocol is $p-persistent$. This probability strongly depends on the network status. We have chosen a different approach since our algorithm tries to avoid, as most as possible, the use of probabilities by introducing a non-probabilistic modification of 802.11, in order to better control the protocol.

Finally, the literature shows that there exists a set of basic scenarios that lead to fairness issues with 802.11 in an ad hoc context [5]. Many of the previously quoted papers are tested on very specific configurations. One of our aim while designing our algorithm is to find a solution for fairness issues in many cases as possible[1].

3 MadMac: A Fair and Efficient Protocol

The approach of MadMac is to provide a schedule on the packets like the one designed in [14] but topology independent and with no extra information than the one provided by 802.11. Of course, a perfect schedule is difficult to obtain with these constraints but the simulation results will show that we obtain good performances.

The basic scheme. The idea behind the proposed protocol comes from the following remarks:

– If an active node senses activity on the channel, then it means that it is not alone on the channel and that at least two stations (including itself) send packets on the radio medium.
– If an active node experiences one or more collisions on its packets, then we can derive the same conclusion: at least two stations (including itself) send packets on the radio medium.

The second statement differs from the first one in the sense that the detected competing stations are not necessarily in communication or in carrier sensing range. However, we can say that, from the point of view of the node that experiences collisions, they share the medium since the station can not successfully send its packets due to interfering transmissions. Note that, considering only the sensing activity and/or the experienced collisions, a node can not deduce how many nodes compete with it. To approximate this number, other operations are required like capturing useful data (the source and the destination for instance) in control and data packets. However it seems difficult to exactly deduce this number as soon as a carrier sensing mechanism is used. Since we don't want to use and send extra information, each node can only deduce, with these two statements, whether it shares the medium (in the general sense) with at least one another node.

If at least one of these two statements is true, then the active node sets a boolean variable, called $SHARE$ to 1. Since the share is not permanent, this variable is updated periodically. We consider a period of $Delta\_Slot$ which the value will be discussed later on. At the beginning of each $Delta\_Slot$, the $SHARE$

[1] All the configurations will not be listed here due to space limitation. See [18] for more details.

variable is reset to 0. The *Delta_Slot* period behaves as a sliding window. When *SHARE* is equal to 1 for one node, this node considers that it shares the medium with one or more stations and reduces its MAC throughput by 2 by introducing a waiting time before each new packet to send. The goal of this waiting time is to introduce an alternate schedule between the competing nodes. This waiting time T_{WAIT} is equal to $T_{DIFS}+M+T_p+T_{SIFS}+T_{ACK}$, where T_p is the packet transmission time of this node, T_{ACK} is the ACK transmission time, T_{SIFS} and T_{DIFS} are respectively SIFS and DIFS duration and M is the mean backoff time of 802.11 (*i.e.* $310\mu s$). T_p can be different for each node but the waiting time introduced allows a full backoff decrementation for the other competing stations. The introduction of this waiting time should increase fairness because the sending is done alternatively. This waiting time is never stopped and is active for each packet that is not entered in the medium access process of 802.11 as soon as *SHARE* becomes equal to 1.

At the end of this waiting time, our algorithm uses the classical medium access algorithm of 802.11 for the packets to send, *i.e DIFS* plus a backoff. Note that a random access can not be removed from our algorithm because *SHARE* only indicates that the medium is shared but the number of competing nodes is unknown. However, since this extra waiting time should reduce collisions, we use a smaller contention window size than 802.11.

If *SHARE* is equal to 0, then our protocol uses the MAC protocol of 802.11. Note that this latter may change the value of *SHARE*, since during the backoff decrement the medium can be sensed busy or the packet can experience one or more collisions. Anyway, even if *SHARE* is set to 1 during this decrement, it is always the MAC protocol of 802.11 that is used to send this packet.

Collision avoidance. To manage collisions, we use the Binary Exponential Backoff algorithm of 802.11, but we keep track of the successive collisions: we use another variable called *NB_COL* that maintains the maximum number of successive collisions encountered by the node in a *Delta_Slot*. This value is set to zero at each new *Delta_Slot*.

If the node senses an activity "and" experiences one or more collisions[2] and $NB\_COL > k$ (k is a parameter of our algorithm), then we consider that the node is very likely in a hidden terminal configuration (see [13] for more details). To avoid the overall throughput decrease due to collisions and the short time unfairness due to the sending of consecutive packets of the same emitter, we force the hidden nodes to emit in turn. For that, as soon as the node succeeds in transmitting the packet that has experienced at least k collisions, then we introduce another waiting time $T_{ALT} = T_{WAIT}+T_{MTU}$ for the following packets, where T_{MTU} is the time needed to transmit a packets of MTU size. The T_{WAIT} part in T_{ALT} is never stopped but the T_{MTU} part can stopped as soon as the node senses activity on the medium (like the ACK from the hidden node). At the end of T_{ALT}, our algorithm uses the classical medium access algorithm of 802.11. Thus, the nodes in competition will alternate their emission. This process

[2] The two statement must be true.

is maintained while an activity is detected. If no activity is detected, then the basic scheme is restarted.

No monopoly on the channel. In some configurations, shown in [5], some nodes may monopolize the radio medium preventing some other stations from accessing to the channel. These nodes never experience collisions and always sense the medium free since the other competing nodes don't succeed in accessing the medium. To solve this problem after x consecutive successful packets sending with $SHARE$ set to 0, the $x + 1$th and $2x + 1$th packet are sent with a larger contention window. This pattern is repeated for the following packets. This process should allow other nodes to access the medium and to send a packet, which will update the $SHARE$ variable of the monopolizing node.

4 Simulation Results

The proposed protocol has been evaluated by simulations using NS-2[3]. The comparison has been performed with 802.11. We have tested most of the basic scenarios presented in [5] and more complex topologies. These studies have been carried out using a constant bit rate application that saturates the medium and a packet size of 1000 kbytes. We have modified some of the NS-2 parameters such as the power and the transmission range to reflect the HR-DSSS 11 Mb/s physical layer of the 802.11b protocol. To avoid message transmission other than those created by the constant bit rate traffic, a static routing agent is used. Other sources of traffic such as those generated by the ARP protocol have also been disabled. Note that the same parameters of MadMac are used in all the presented simulations.

Performance of one-hop networks. The first simulations have been performed on the simple scenarios where communications take place between nodes that are in communication range of each other. In these scenarios there is no fairness issue and the goal is to compare the global throughput of MadMac with 802.11 in this classical configuration. The results given on Fig. 1 show that our protocol provides a higher overall throughput than 802.11. This is due to the fact that the contention window size is set to a lower value than in 802.11.

The achieved global throughput with two active nodes is also higher than with 802.11, but is smaller than with one or three active nodes. This is due to the fact that the two nodes alternate their emissions and this alternation is almost perfect. Therefore the overlapping of the backoff decrement is rare in this configuration. For scenarios with more than two stations, the last emitter is in the waiting phase while the other nodes finish their waiting phase or enter in the 802.11's process, *i.e.* the backoff decrement (after a $DIFS$). Therefore, there is an overlapping of the backoff decrement phases, which leads to a smaller time interval between two consecutive packets sent on the medium than with two nodes. Here the backoff process of 802.11 guarantee fairness on channel access.

[3] Network Simulator http://www.isi.edu/nsnam/ns/

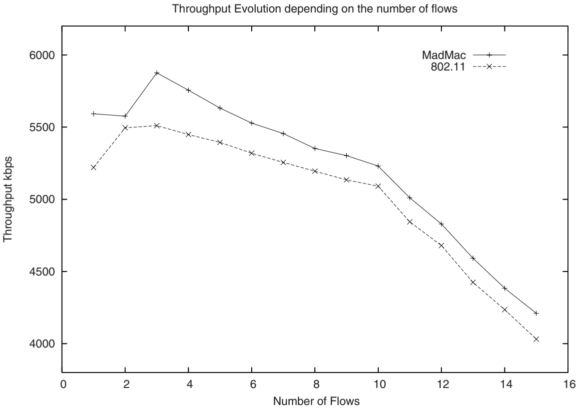


Fig. 1. Total throughput depending on the number of active nodes in an ad hoc cell

We see also that the overall throughput of MadMac decreases with the number of contending nodes (like for 802.11), but is always higher than 802.11. This decreasing is due to the increase of collisions for the two protocols. As the contention window size of MadMac is smaller than the one of 802.11, the number of collisions with MadMac is a little bit higher. But we see that it does not drastically reduce the throughput and MadMac is efficient.

Henceforth, we consider that the radio medium capacity, denoted C, obtained with MadMac (802.11 resp.), is the throughput achieved with one emitter and corresponds to 5.6 Mb/s (5.2 Mb/s resp.). We will use this value in the following to derive a metric for efficiency of the tested scenarios, as explained in the following.

Metrics. In [14], the authors investigate the trade-off between aggregate throughput and fairness. They show the fundamental conflict between achieving flow fairness and maximizing overall throughput: if a fairness scheme is adopted on flow rates then it may be impossible, for some configurations to maximize aggregate throughput. Then, we think that the maximum aggregate throughput (called also capacity) is not an adapted metric to evaluate the efficiency of a fair protocol. Instead, we use as a metric of efficiency the aggregate throughput that is achieved when the flow rates are allocated according to a fairness scheme. Henceafter, we call this aggregate throughput *fair capacity*. Note that the fair capacity depends on a fairness scheme. Like many articles that deal with fairness in ad hoc networks, we have considered the max-min fairness scheme, as it is considered as the fairer scheme[4]. To evaluate the fairness of our solution, we use the fairness index defined in [12]. Since we base our evaluation on the max-min fairness, the fairness index is the following: $\frac{(\sum_i r_i/r_i^*)^2}{n\sum_i (r_i/r_i^*)^2}$, where r_i is the

[4] But not as the most efficient in terms of global performance. The discussions on the quality of the max-min fairness in the ad hoc context are out of the scope of this article.

rate achieved by our solution on flow i, r_i^* is the rate on flow i in the max-min fairness allocation and n is the number of flows.

All the tables in the following, unless specified, give the aggregate throughput (in kb/s and with their confidence interval) and the max-min fairness index achieved with 802.11 with or without RTS/CTS and with MadMac.

The hidden terminal configuration. One tested scenario is the well-known hidden terminal problem depicted in Fig. 2. In this scenario, nodes 1 and 2 are fully independent. The main problem with 802.11 is the high number of collisions, which leads to an increase of the contention window size that drastically reduces the throughput of nodes 1 and 2. The RTS/CTS mechanism has been proposed to increase the throughput but this solution is not so efficient and introduces a short term fairness issue.

From Table 1, we see that these two protocols are fair compared to a max-min fairness allocation, but MadMac is much more efficient than 802.11 since the overall throughput of MadMac is much higher than the one achieved by 802.11. Moreover the aggregate throughput of MadMac is very close to the fair capacity under a max-min fairness scheme. The fair capacity in this configuration is equal to C and corresponds to 5.6 Mb/s. This is due to the fact that, with MadMac, the hidden nodes almost perfectly alternate their emission, which does not result in many collisions. Therefore their contention window size remains low and the difference between these two sizes is also low. We can notice the short time unfairness induced by the RTS/CTS mechanism (the confidence interval is large.). An appropriate and simple scheduling, as the one achieved in MadMac, can solve the hidden terminal problem.

Another impact of the hidden terminal configuration. In the third scenario, we propose to study another impact of the hidden terminal scenario, depicted in Fig. 3. This configuration has first been pointed out in [3]: node 1 (3 resp.) sends data to node 2 (4 resp.) and nodes 2 and 3 are in communication range,

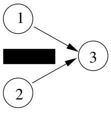


Fig. 2. Hidden terminal

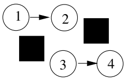


Fig. 3. Another hidden terminal

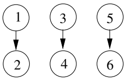


Fig. 4. 3 pairs

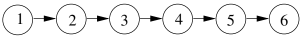


Fig. 5. Chain

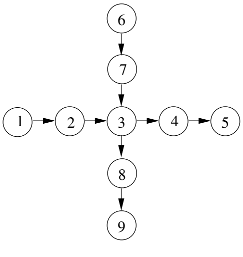


Fig. 6. Star

Table 1. Results on hidden terminal scenario

		Th. kbps	Conf. Int. (0.05)
802.11	total	3640.84	3636.64 - 3645.04
	index	0.9999	
802.11 RTS/CTS	total	3882.68	3870.83 - 3894.53
	index	0.9999	
MadMac	total	5561.32	5559.49 - 5563.15
	index	1.0000	

Table 2. Another impact of the hidden terminal scenario: results

		Th. kbps	Conf. Int. (0.05)
802.11	total	5217.31	5212.41 - 5222.21
	index	0.5000	
802.11 RTS/CTS	total	3964.56	3959.01 - 3970.10
	index	0.5808	
MadMac	total	4452.04	4442.26 - 4461.83
	index	0.9364	

whereas node 1 (4 resp.) is independent of nodes 3 and 4 (1 and 2 resp.). In this scenario, node 3's transmission always succeeds, whereas node 1's transmission experiences collision. The only chance for node 1 to successfully transmit a packet is when its frame is sent during a silent period of node 3. The use of RTS/CTS mechanism can reduce the number of collisions in 2 because the length of the RTS frames is often smaller than the data frames, but this use is not very efficient (see Tab. 2).

From Table 2, we see that MadMac is fairer than 802.11 with or without RTS/CTS. This is due to the introduction of T_{WAIT} by the pair $3-4$, which leads to more successful transmissions for node 1. However, the overall throughput of MadMac is smaller than the fair capacity equal to C (and corresponding to 5.6 Mb/s), even if it is higher than the one of 802.11 with RTS/CTS. This difference is due to the fact that collisions still exist since the alternation is not perfect between the two emitters and since every *Delta_Slot* the two sources reset their *SHARE* variable, which leads to a direct emission of the packets without extra waiting time. Note that MadMac does not consider this configuration as a hidden node scenario since node 1 never detects activity on the medium even if it experiences collisions.

These simulations show, once more, that it is possible to replace the RTS/CTS mechanism by an appropriate MAC scheme that is more efficient and fairer.

The three pairs. The fourth studied scenario is the three pairs scenario depicted in Fig. 4 and pointed out in [6]. In this scenario, nodes 1 and 5 are fully independent and node 3 is in the carrier sensing range of nodes 1 and 5. With 802.11, the backoff decrement of node 3 can only take place when nodes 1 and 5 are in their silence period. As these two nodes are not synchronized, the silence period for node 3 is rare and the probability to node 3 to access the medium is low.

From Table 3, we see that MadMac is much fairer than 802.11. On the other hand, MadMac is less efficient than 802.11, but its overall throughput is very close to the fair capacity equal to $\frac{3C}{2}$ (and corresponding to 8.4 Mb/s). We have here a typical example of trade-off between efficiency and fairness.

The performance anomaly. This well-known scenario presents a fairness issue due to different throughputs on the network (see [2]). In this scenario, nodes in

Table 3. 3 pairs scenario: Results

		Th. kb/s	Conf. Int. (0.05)
802.11	total	10331.18	10309.71-10352.66
	index	0.6842	
MadMac	total	8308.90	8308.20 - 8309.59
	index	0.9999	

Table 4. Performance anomaly: Results

		Th. kb/s	Conf. Int. (0.05)
802.11	11Mb	1231.74	1212.54 - 1250.94
	2Mb	1236.13	1227.64 - 1244.62
	total	2467.87	2453.47 - 2482.27
MadMac	11Mb	1674.06	1673.97 - 1674.14
	2Mb	837.12	837.07 - 837.18
	total	2511.18	2511.07 - 2511.29

communication range are trying to send their frames at different data rate. The node sending at the lowest rate reduces the throughput of all the nodes transmitting at higher data rate to a value close to the throughput of the slowest node.

Simulations have been performed with frames of 1000 bytes and with two nodes transmitting at 2 Mb/s and 11 Mb/s. This scenario is different from the previous ones since the flow rates are different and a solution to this issue rather seeks for a time fairness. Therefore, we only investigate, here, the efficiency and the rate of each flow. From Table 4, we can see that MadMac provides a better time sharing of the medium and slightly increases the overall throughput. This is due to the fact that the waiting time introduced by MadMac is equal to the time transmission of the packet. Thus, the waiting time for a node transmitting at a low data rate is greater than for the node transmitting at a high data rate. This difference between the waiting times allows a node with smaller waiting time to send more packets.

Other simulations. We have evaluated more complex topologies. Due to space limitation, we only give the results of two scenarios, depicted on Figures 5 and 6. They are interesting because they combine different issues with the presence of multiple basic configurations (as the ones of Figures 2, 3 and 4).

Table 5. Results on Chain

		Th. kbps	Conf. Int. (0.05)
802.11	total	9411.77	9374.68 - 9448.86
	index	0.6511	
802.11	total	7201.53	7171.10 - 7231.96
RTS/CTS	index	0.6827	
MadMac	total	8339.70	8242.65 - 8436.74
	index	0.7995	

Table 6. Results on Star

		Th. kbps	Conf. Int. (0.05)
802.11	total	19196.92	19126.95 - 19266.89
	index	0.5139	
802.11	total	14698.49	14653.49 - 14742.71
RTS/CTS	index	0.5047	
MadMac	total	8013.34	7900.95 - 8125.72
	index	0.7089	

From Table 5, we see that MadMac is fairer than 802.11 with and without RTS/CTS, and less efficient than 802.11 without RTS/CTS, but more efficient than 802.11 with RTS/CTS. Once more, our solution gives more good results than 802.11 with RTS/CTS, since MadMac achieves a better fairness and a better overall throughput that is not so far from the fair capacity equal to $\frac{5C}{3}$ (corresponding to 9.3 Mb/s).

From Table 6, we see that MadMac is much fairer than 802.11 but less efficient. This topology is a typical example where the trade-off is very difficult to find because MadMac achieves a high aggregate throughput compared to the fair capacity equal to C (and corresponding to 5.6 Mb/s). The fairness is difficult to obtain and some flows are penalized, like the flows between nodes 7 and 3 and nodes 2 and 3. These flows are in a configuration that combines multiple issues (the hidden station problem, two problems of Figure 3 and the three pairs problem).

5 Conclusion

In this paper, we have proposed a new MAC protocol based on 802.11, called MadMac, that provides more fairness than 802.11 while maintaining a good aggregated throughput in the network. We have compared MadMac with 802.11 from fairness and efficiency points of view. These comparisons have been carried out in many basic scenarios that are known to lead to fairness issues and in more complex topologies. Results, from these simulations, show that, in most of the cases, MadMac is close to the fair capacity while ensuring fairness among the flows.

MadMac is based on several parameters that can be fine tuned to improve its performances. We have started to study these parameters, like for instance, the values to give to the parameters Delta Slot, k, x and the use of different packet sizes [18]. The obtained results are very promising, but the main problem with fine tuning these parameters is that the modification of one parameter value can improve the protocol performance on only specific scenarios while the performance decreases on other configurations. A very careful analysis should have to be carried out to select the best values to give to the parameters, i.e. the values that will lead to the better performances in most of the cases.

Future works would be to investigate other ad hoc topologies like random topologies. We also plan to compare MadMac to other fair protocols such as PNAV [4] or EHATDMA [9].

Our initial assumptions are very restricting since MadMac considers very limited information (the carrier sensing and the number of collisions). The fairness and the efficiency of our protocol can clearly be enhanced with extra information. In the future, we plan to add in MadMac information from other layers of OSI model such as neighbors table from routing layer for instance, in order to measure the impact of such information on the performances.

References

1. B. Bensaou, Y. Wang, and C. C. Ko. Fair medium access in 802.11 based wireless ad-hoc networks. In *MobiHoc*, pages 99–106, Piscataway, NJ, USA, 2000. IEEE Press.
2. G. Berger-Sabbatel, F. Rousseau, M. Heusse, and A. Duda. Performance anomaly of 802.11b. In *INFOCOM*, 2003.

3. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. Macaw: a media access protocol for wireless lan's. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 212–225, New York, NY, USA, 1994. ACM Press.
4. C. Chaudet, G. Chelius, H. Meunier, and D. Simplot-Ryl. Adaptive probabilistic nav to increase fairness in ad hoc 802.11 mac layer. In *MedHoc NET*, 2005.
5. C. Chaudet, D. Dhoutaut, and I. Guérin-Lassous. Performance issues with ieee 802.11 in ad hoc networking. *IEEE Communication Magazine*, 43(7), July 2005.
6. D. Dhoutaut and I. Guérin Lassous. Impact of Heavy Traffic Beyond Communication Range in Multi-Hops Ad Hoc Networks. In *INC*, Plymouth, Royaume-Uni, July 2002.
7. Z. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks. In *INFOCOM*, 2004.
8. IEEE Standard for Information Technology Telecommunications and Information Exchange between Systems. Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.
9. J. He and H.K. Pung. Fairness properties of medium access control protocols for multi-hop ad hoc wireless networks. *Elsevier publication*, to appear, 2005.
10. M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans. In *SIGCOMM '05*, pages 121–132, New York, NY, USA, 2005. ACM Press.
11. X. L. Huang and B. Bensaou. On max-min fairness and scheduling in wireless adhoc networks: analytical framework and implementation. In *MobiHoc '01*, pages 221–231, New York, NY, USA, 2001. ACM Press.
12. R. Jain. Throughput fairness index: An explanation, 1999.
13. Z. Li, S. Nandi, and A. K. Gupta. Modeling the short-term unfairness of ieee 802.11 in presence of hidden terminals. In *NETWORKING*, pages 613–625, 2004.
14. H. Luo, S. Lu, and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. In *MobiCom '00*, pages 76–86, New York, NY, USA, 2000. ACM Press.
15. T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving mac layer fairness in wireless packet networks. In *MobiCom '00*, pages 87–98, New York, NY, USA, 2000. ACM Press.
16. T. Ozugur, M. Naghsineh, P. Kermani, and J. A. Copeland. Fair media access for wireless lans. In *GlobeCom*, Rio de Janeiro, Brazil, 1999.
17. D. Qiao and K. Shin. Achieving efficient channel utilization and weighted fairness for data communications in ieee wlan under the dcf. In *IEEE Int'l Workshop on QoS*, pages pp.227–36., 2002.
18. T. Razafindralambo and I. Guérin-Lassous. Increasing fairness and capacity using madmac protocol in 802.11-based ad hoc networks. Technical report, INRIA, 2005.
19. Y. Wang and B. Bensaou. Achieving fairness in IEEE 802.11 DFWMAC with variable packet lengths. In *GlobeCom*, 2001.

Duplicate Address Detection in Wireless Ad Hoc Networks Using Wireless Nature

Yu Chen and Eric Fleury

ARES/INRIA – INSA de Lyon, France
Eric.Fleury@inria.fr, ychen@cs.tamu.edu

Abstract. We consider duplicate address detection in wireless ad hoc networks under the assumption that addresses are unique in two hops neighborhood. Our approaches are based on the concepts of *physical neighborhood views*, that is, the information of physically connected nodes, and *logical neighborhood views*, which are built on neighborhood information propagated in networks. Since neighborhood information is identified by addresses, inconsistency of these two views might occur due to duplicate addresses. It is obvious that consistency of these two views on each node's neighborhood is necessary for a network to have unique addresses, while the sufficiency depends on the types of information contained in neighborhood views. We investigate different definitions of neighborhood views and show that the traditional neighborhood information, neighboring addresses, is not sufficient for duplication detection, while the wireless nature of ad hoc networks provides useful neighborhood information.

Keywords: duplicate address detection, wireless ad hoc networks, symmetry.

1 Introduction

A wireless ad hoc network is a set of wireless nodes which cooperatively and spontaneously form a network. Such a network provides a flexible means of communication without using any existing infrastructure or centralized administration. Significant research in ad hoc networks has focused on routing, the majority of which assume that nodes are configured *a priori* with a unique address. Since in ad hoc networks nodes join and leave at will, automated address assignment is required to dynamically configure nodes. In traditional networks, dynamic address assignment can be performed by a DHCP server [8]. But this solution is not suitable in ad hoc networks due to the unavailability of centralized servers. One alternative is to allow nodes to pick tentative addresses and the uniqueness of picked addresses is checked by some duplication detection mechanism; new tentative addresses are picked if duplications are detected [3, 17, 18, 19].

In this work, we focus on *duplicate address detection* in wireless ad hoc networks. Works on duplication detection have been proposed previously (e.g., [3, 17, 18, 19]). Many approaches assume the existence of global unique identification. Under this assumption, duplication can be detected by propagating

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 415–426, 2006.

associations of identifications and addresses. However, no global identification is truly unique; e.g., IEEE medium access control (MAC) addresses are not truly unique. One alternative is to create an identification randomly. The argument is that the probability of collision is small if the range of identifications is large enough. But propagating large-ranged identifications will cause large packet overhead. Thus relying on global uniqueness is not desirable. In our work, we consider detecting duplicate address based on *local* uniqueness: addresses are assume to be *unique in two hops neighborhood*. This assumption is made due to two facts. First, symmetry can prevent a problem to be solved in anonymous networks [2, 9, 10], thus some form of uniqueness is necessary; compared to the assumption of *global* unique identifications, our assumption is much weaker. Second, many algorithms have been proposed to assign addresses that are unique in two hops neighborhood (e.g., [11]).

We observe that protocols that are not aware of duplicate addresses behave as if all the packets from the same address are from the same node. For example, link state routing running on a node with address *ip* regards all the nodes that are connected to a node with address *ip* as its neighbors. Thus if duplicate address exists, the view of link state routing on the neighborhood is different from the physical neighborhood view. Based on this observation, we propose the concepts of *physical neighborhood views* and *logical neighborhood views*. Informally, a *physical neighborhood view* of a node is information of nodes physically connected to it; examples include the number of neighbors, addresses of neighbors and distances to each neighbor. A *logical neighborhood view* is built based on neighborhood information identified by *addresses*: a node with address *ip* considers all the nodes that connect to a node that has address *ip* as its neighbors and the view is built based on neighborhood information of all such "neighbors". For example, given a node that has address *ip*, the number of its neighbors in its physical view is the number of nodes *physically connected to it*, and in its logical view it is the number of nodes *connected to a node that has address ip*. More detailed example will be given later in Figure 1.

We consider duplicate address detection by comparing the physical and logical neighborhood views of each node. Logical neighborhood views can be built if each node propagates to all the others the state of each of its links, identified by ends' addresses. Since neighborhood information is required by most existing protocols and it usually contains two ends' addresses of each link, the overhead of our approaches depends on other information defined in neighborhood views. It is obvious that consistency of physical and logical views on each node's neighborhood is necessary for a network to have unique addresses, but whether it is sufficient depends on the types of information contained in neighborhood views. For example, if a neighborhood view is defined as the number of neighbors, it is sufficient only in a small class of networks.

We investigate different definitions of neighborhood views. We start from a traditional definition of neighborhood views, which consists of neighboring addresses. The idea of detecting duplication by comparing neighboring addresses has been proposed in PDAD-NH [19, 18]. But no further investigation on the correctness is given. It is claimed "in case the sender of the link state packet is a common neighbor of the nodes with the same address, the conflict cannot be detected by PDAD-NH. Thus, conflicts in the two hops neighborhood must again be detected by other means". We take a close look at this approach under the assumption of unique address in two hops neighborhood and prove it fails in certain class of networks. We show this class of networks have the following properties: each existing address is assigned to the same number of nodes and there is a circle that has special properties. This class of networks might not be common in practice, but should not, therefore, be overlooked, since its existence indicates an important difference between wired and wireless networks. The properties of this class of networks provide strong hints for our second definition of neighborhood views, which also includes distance in x and y direction to each neighbor. We show that, under the assumption of unique addresses in two hops neighborhood, duplication can be detected if distance information satisfies certain accuracy, which means distance information can be represented in a small number of bits and overhead can be small. Note we do not assume the availability of strong position information such as GPS. Relative distance between neighboring nodes can be estimated by the signal strength or microwave or more sophisiticated techniques like microwave [14, 20, 1]. Neighbor or stronger distance information is used in many works on wireless networks [4, 14, 15].

2 Related Work

Dynamic Host Configuration Protocol (DHCP) [8] is commonly used for dynamic address assignment in traditional networks. Works on dynamic address assignment for ad hoc network include [12, 16, 13]. Stateless approaches for local networks are proposed in [12] and [16], which require all nodes to be reachable in one-hop. In [13], addresses are treated as a shared resource and it is managed by a distributed mutual exclusion.

Solutions for duplication detection in ad hoc networks has been proposed previously (e.g. [3, 17, 18, 19]). In [3], each node has an fixed-length identifier which is randomly generated. A special message that includes nodes' address and identifier is diffused to the entire network; a node detects a duplicate address when it receives a message that has the same address as its own, but with a different identifier. Global unique or randomly generated keys are assumed in [17], in which duplication is detected by attaching key information in link state packets. The approach proposed in [17] successfully prevents packets from being delivered to wrong destinations. Most approaches for duplicate address detection require propagation of key information, which causes high packet overhead. Since lower protocol overhead is one of the most important design goals for wireless ad hoc networks, works have been done in achieving efficiency in terms of protocol overhead. Protocols proposed in [19] and [18] generate almost no protocol overhead: it detects address conflicts in a passive manner based on anomalies in routing protocol traffic. In particular, the idea of detecting duplication by comparing neighborhood information is proposed in approach PDAD-NH [19] [18].

However, no correctness proof is presented. In our work, we show this approach works in most networks, except a special class of networks; the existence of this class of networks indicates the different ability of wired and wireless networks in duplication detection using neighborhood information.

Much work has been done on anonymous networks in which no identifications are available [2, 9, 10, 7, 6]. Less work considers networks, especially wireless networks, with partial identifications. However, partial identification information, such as MAC addresses, are commonly available. Here we consider duplication detection using neighborhood information under the assumption of local uniqueness, which is not solvable in typical wired networks, but can be solved in ad hoc networks by using information provided by wireless nature.

3 System Model and Overview

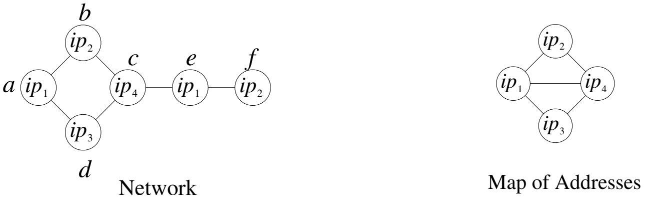
We focus on stand-alone wireless ad hoc networks in which wireless nodes do not have access to a centralized server that could assign network-wide unique addresses. Instead of assuming global unique identifications, we consider duplication detection under the assumption that addresses are unique in two hops neighborhood. In our work, duplicate address is detected by each node comparing its *physical neighborhood view* and *logical neighborhood view*. In section 1, we have given an informal description of physical and logical neighborhood views. In the sequel, we focus on whether the consistency of physical neighborhood view and logical neighborhood view on every node is sufficient for a network to have unique addresses. If it is sufficient, in a network that has duplicate address, at least one node will detect duplication and it can inform other nodes. In our work, we examine two definitions; each definition has its own assumptions on neighborhood knowledge.

Physical neighborhood views are built based on neighborhood knowledge that are assumed to be available, thus no packet overhead is caused. But building logical neighborhood views requires propagation of neighborhood information, which causes packet overhead. We assume each node that has address *ip* generates packets $\langle ip, ip', link\_state\rangle$ for each neighbor that has address ip'; the field *link_state* will be specified by the specific approach. We borrow the name from link state routing and call these packets as *link state packets*. Since neighborhood information is required by most protocols and how to propagate this information is out of the scope of this paper, we assume each node receives link state packets from all the other nodes without going into details of how these packets are propagated. Since most neighborhood information contains two ends' addresses of each link, we evaluate the overhead of each approach based on the packet complexity of field *link_state* in link state packets.

In the first approach (section 4), we assume neighboring addresses are available and neighborhood view is defined as a set of neighboring addresses. No overhead is introduced. We prove this information is not sufficient and this approach fails in certain class of networks; in this class of networks, all the existing addresses are assigned to the same number of nodes and there is a circle in which the sequence of nodes' addresses consists of repeated patterns. Based on this property, we propose our second definition (section 5). We observe that, due to its wireless nature, neighbor distance information is available in ad hoc networks. In our second approach, neighborhood view is defined as distances in x and y direction to each neighbor, together with ends' addresses of each link. Overhead of this approach is distant information in link state packets. We show that duplication can be detected if nodes that have the same address are not too "close"; the meaning of "being close" depends on the accuracy of neighbor distance. The allowance of inaccuracy implies a small number of bits can be used to represent distance information. Based on the assumption that addresses are unique in two hops neighborhood, only small overhead is required.

Before we present our definitions of neighborhood views, we introduce the concept of *addresses map*, which is used in our analysis. Informally, addresses map is a view of a network in which all the nodes with the same address are combined into one. An example is given in Figure 1.

Definition 1. ***(Addresses Map)*** *Given a network G, its addresses map is a graph such that each vertex is a distinct existing address and there is an edge between addresses ip_1 and ip_2 iff there exits link state packet $\langle ip_1, ip_2\rangle$ or $\langle ip_2, ip_1\rangle$.*



node	links identified by ends' addresses
a	$\langle ip_1, ip_2\rangle$, $\langle ip_1, ip_3\rangle$
b	$\langle ip_2, ip_1\rangle$, $\langle ip_2, ip_4\rangle$
c	$\langle ip_4, ip_1\rangle$, $\langle ip_4, ip_2\rangle$, $\langle ip_4, ip_3\rangle$
d	$\langle ip_3, ip_1\rangle$, $\langle ip_1, ip_4\rangle$
e	$\langle ip_1, ip_4\rangle$, $\langle ip_1, ip_2\rangle$
f	$\langle ip_2, ip_1\rangle$

address ip	neighbors of ip in the map
ip_1	ip_2, ip_3, ip_4
ip_2	ip_1, ip_4
ip_3	ip_1, ip_4
ip_4	ip_1, ip_2, ip_3

Fig. 1. An Example of Addresses Map

We use terms "addresses" and "edges" to refer to vertices and links in the addresses map respectively, and "nodes" and "links" to refer to vertices and links in a network respectively. We say a link connects two addresses ip and ip' if its two ends have addresses ip and ip'. The lemma below shows a *necessary* condition for a network to have duplicate address. Proof is not provided due to limited space; it can be found in full paper[5]. Note the existence of circles in its addresses map is not a *sufficient* condition for duplication to exist in a network.

Lemma 1. *Given a network in which addresses are unique in two hops neighborhood, if no circle exists in its addresses map, then no duplicate address exists.*

4 Duplication Detection Using Neighboring Addresses

In this section, the only assumption on neighborhood knowledge is that each node knows its neighboring addresses. The view of a node's neighborhood is defined as *the set of neighboring addresses.* Since no information except ends' addresses of each link is required to build logical neighborhood views, link state packets have form of $\langle ip, ip' \rangle$ and no overhead is caused.

Definition 2. *Given a network and a node n that has address ip, we define*

- *physical neighborhood view of n $\equiv$ the set of addresses of nodes that are physically connected to n.*
- *logical neighborhood view of $n \equiv \{ip' | \exists$ link state packet $\langle ip, ip' \rangle\}$*

The term "view" is used in this section according to this definition. Table 1 describes physical and logical neighborhood views of the network in Figure 1, in which this approach works. However, special symmetry can prevent this approach from detecting duplications. Counterexamples are given in Figure 2: all the nodes in Network 1 and Network 2 have consistent views, but duplications exist in both networks. Note these two networks have the same addresses map.

Table 1. An Example: Views of Neighborhood of Network in Figure 1

node	physical view	logical view	Consist.	node	physical view	logical view	Consist.
a	ip_2, ip_3	ip_2, ip_3, ip_4	False	b	ip_1, ip_4	ip_1, ip_4	True
c	ip_1, ip_2, ip_3	ip_1, ip_2, ip_3	True	d	ip_1, ip_4	ip_1, ip_4	True
e	ip_2,ip_4	ip_2, ip_3, ip_4	False	f	ip_1	ip_1, ip_4	False

Now we investigate the properties of networks in which this approach fails. The lemma below shows that in such a network, addresses are distinct in the shortest path connecting any two different addresses. Due to limited space, proof is not provided and it can be found in [5].

Lemma 2. *Consider a network in which addresses are unique in two hops neighborhood and views are consistent on every node. Given any two addresses ip_x and ip_y, $ip_x \neq ip_y$, nodes in a shortest path that connects ip_x and ip_y have distinct addresses.*

The lemma below states that in such networks, given a path in which nodes have distinct addresses, there are t distinct paths that have same sequence of addresses, where t depends on the number of nodes that have the same address.

Lemma 3. *Consider a network in which addresses are unique in two hops neighborhood and views are consistent on every node. Given a path $path_0$ in which all the nodes have distinct addresses. Let t be the number of nodes that have address ip_0, where ip_0 is the address of the first node in $path_0$. Then there exist t distinct paths that have the same sequence of addresses as $path_0$.*

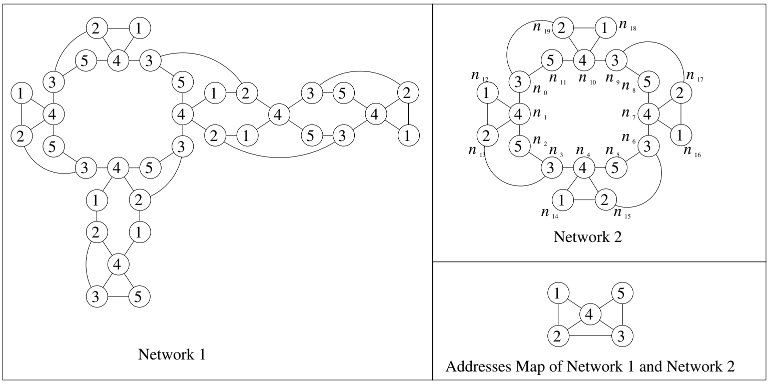


Fig. 2. Examples in which views are consistent & duplicate addresses exist

Proof. Let $path_0$ be $\langle n_0^0, n_0^1, \ldots, n_0^k \rangle$. We denote the address of n_0^i by ip_i. We construct t paths by induction. For some $s \in [0, t-2]$, we assume there are $s+1$ paths, $path_0$, …, $path_s$, that are distinct and have the same sequence of addresses as $path_0$. Note it is true when $s = 0$. We construct $path_{s+1}$ as follows.

We denote nodes in $path_0, \ldots, path_s$ by $paths_{[0,s]}$. Since there are t nodes that have address ip_0 and only $s + 1 \leq t - 1$ of them are in $paths_{[0,s]}$, there is at least one node that has address ip_0 and is not in $paths_{[0,s]}$. Let this node be n_{s+1}^0.

Now we construct the rest of this path by induction. For $i \in [0, k-1]$, assume there is a path $\langle n_{s+1}^0, \ldots, n_{s+1}^i \rangle$ such that: (1) $\forall l \in [0, i]$, the address of n_{s+1}^l is ip_l; and (2) all the selected nodes, that is, $\{n_{s+1}^0, \ldots, n_{s+1}^i\} \bigcup paths_{[0,s]}$, are distinct. We select n_{s+1}^{i+1} as follows (Figure 3). Since views are consistent on every node and n_{s+1}^i and n_0^i have the same address, n_{s+1}^i and n_0^i have the same set of neighboring addresses. Since n_0^i has a neighbor n_0^{i+1} that has address ip_{i+1}, n_{s+1}^i also has a neighbor that has address ip_{i+1}, denoted by n'. Now we show n' is not among the selected nodes. Among all the selected nodes, only nodes in $\{n_0^{i+1}, \ldots, n_s^{i+1}\}$ have address ip_{i+1}. If n' is one of them, then n' is connected to n_l^i for some $l \in [0, s]$. So n' is connected to n_l^i and n_{s+1}^i, which have the same address ip_i. It contradicts to the assumption of unique addresses in two hops neighborhood. Since n' has not been selected and it has address ip_{i+1}, it can be selected as n_{s+1}^{i+1} and the lemma is proved.

Based on these two lemmas, the following theorem states that all the existing addresses are assigned to the same number of nodes. An interesting implication is that a network with a prime number of nodes does not have duplicate address if views are consistent on every node. We define the *duplicate degree* of such a network as the number of nodes that take the same existing address.

Theorem 1. *Consider a network in which addresses are unique in two hops neighborhood and views are consistent on every node. Given an address ip, denoting the number of nodes that take ip as its address by $f(ip)$, we have $f(ip') = f(ip'')$ for any two addresses ip' and ip'' that exist in the network.*

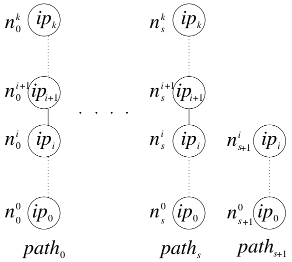


Fig. 3. Proof of Lemma 3

Proof. Assume in contradiction that there exist addresses ip_x and ip_y such that the number of nodes that have address ip_x is s and the number of nodes that have address ip_y is t, where $s, t \geq 1$ and $s > t$. Consider all the pairs x' and y' such that x' has address ip_x and y' has address ip_y. Let x and y be the closest pair among all these pairs. Let $path_0$ be the shortest path between x and y. By Lemma 2, the addresses of nodes in $path_0$ are distinct. By lemma 3, there are s paths with the same sequence of addresses as $path_0$ and all the nodes in these paths are distinct. So there are s nodes that have the same address as y, which contradicts to that only t, $t < s$, nodes has address ip_y.

The above theorem examines the connection between the *number* of nodes and that of addresses. Now we take a close look at the connection between the *topology* of a network and that of its addresses map. In particular, given a subgraph S_A of its addresses map, we examine the subgraph of a network that is "relevant" to S_A. Informally, a node is relevant if it has an address in S_A and a link is relevant if an edge connecting its two ends' addresses exists in S_A. We say such a subgraph is expanded by S_A. The formal definition is given below.

Definition 3. ***(Expanded Subgraph):*** *Given a network G and a subgraph S_A of its addresses map, we consider a subgraph S_G of G that satisfies: nodes in S_G are the nodes that have addresses in S_A, and there is an link between nodes x and y in S_G iff there is an edge between the address of x and the address of y in S_A. We say S_G is the subgraph that is expanded by S_A.*

In Theorem 2, we examine addresses that are organized in a circle in the addresses map. We show that the subnetwork expanded by it consists of a set of circles. Furthermore, if duplication exists, there is a "minimal" circle in the addresses map which expands a subgraph that contains a circle with duplicate addresses; the existence of such a circle provides strong hints for our second approach. A "minimal circle" is defined below. For example, in Network 2 of Figure 2, circle $\langle 3, 4, 5, 3 \rangle$ is minimal while circle $\langle 1, 2, 3, 4, 1 \rangle$ is non-minimal.

Definition 4. ***(Minimal Circle):*** *Given a graph G, a circle cir is minimal iff there exists a node x in cir such that cir is the shortest circle that contains x.*

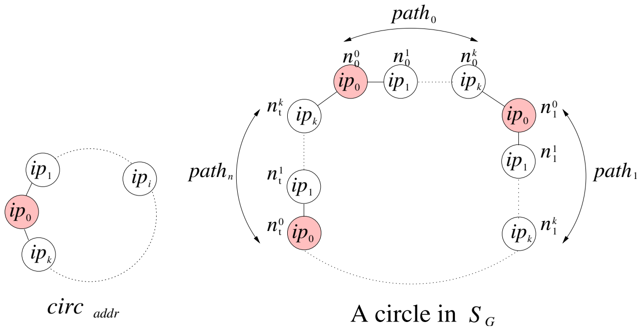


Fig. 4. Theorem 2

Theorem 2. *Consider a network G in which addresses are unique in two hops neighborhood and views are consistent on every node. Given any circle $circ_{addr} = \langle ip_0, ip_1, \ldots, ip_k, ip_0 \rangle$ in the addresses map, the subgraph S_G of G that is expanded by $circ_{addr}$ consists of a set of circles, and each circle has the form of*

$$path_0 \circ path_1 \cdots \circ path_{s-1} \circ \langle n_0 \rangle$$

where $path_i$ is a path that has sequence of addresses $\langle ip_0, \ldots, ip_k \rangle$, n_0 is the first node in $path_0$ and $s \geq 1$ (Figure 4).

Furthermore, if duplicate address exists, there exists a minimal circle in the addresses map whose expanded subgraph in G contains a circle that has $s > 1$ in the above form.

Proof is not provided due to limited space; it can be found in [5]. We consider Network 2 in Figure 2 as an example. The duplicate degree is 4. The subgraph expanded by the circle of addresses $\langle 1, 2, 4, 1 \rangle$ consists of four circles. The subgraph expanded by a minimal circle $\langle 3, 4, 5, 3 \rangle$ is one circle with $s = 4$: $\langle n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9, n_{10}, n_{11} \rangle$. A non-minimal circle $\langle 1, 2, 3, 4, 1 \rangle$ also expands a subgraph that has $s = 4$: $\langle n_{12}, n_{13}, n_3, n_4, n_{14}, n_{15}, n_6, n_7, n_{16}, n_{17}, n_9, n_{10}, n_{18}, n_{19}, n_0, n_1, n_{12} \rangle$.

5 Duplication Detection Using Neighbor Distance

In Theorem 2, we show that if duplication exists and views defined as neighboring addresses are consistent at every node, there exists in the network a circle which consists of pattens of nodes that have the same sequence of addresses. For example, a pattern in Figure 4 is $\langle n_i^0, \ldots, n_i^k, n_{i+1}^0 \rangle$ (here we write patterns in such a way that the first node of the next pattern is the last node of the last pattern). In order to form a circle, either the distance between two ends of each pattern is zero, which means two nodes with address ip_0 are at the same location; or patterns do not have the same shapes and orientations, since otherwise the end of the last pattern cannot go back to the beginning of the first pattern. Since the sequence of addresses is the same for all the patterns, difference in shapes and orientations means relative distances between neighbors differ at nodes with the same address. Thus if neighbor distance information is included, inconsistency of neighborhood views will be detected. Since in practice accurate distance information might not be available due to inaccuracy in measurement or limitation in the number of bits to represent distance information, we consider duplication detection using inaccurate distance information with bounded error, instead of accurate information. Note in this approach, the only modification of the original link state routing is to attach relative distances to neighbors in link state packets.

We denote the real x-coordinate (y-coordinate resp.) of node n by $x_{coor}(n)$ ($y_{coor}(n)$ resp.), and the real distance from node n to node n' in x-direction (y-direction resp.) by $dis_X(n,n')$ ($dis_Y(n,n')$ resp.). We assume each node n has distance information to each neighbor n' in x-direction and y-direction, denoted by $dis_{X\_inf}(n,n')$ and $dis_{Y\_inf}(n,n')$ respectively. Node n that has address ip generates link state packet for each of its neighbor n' that has address ip' in the form of $\langle ip, ip', d_x, d_y\rangle$, where $d_x = dis_{X\_inf}(n,n')$ and $d_y = dis_{Y\_inf}(n,n')$. Note distance information obtained by each node might differ from the real information. Let e_{rr} be the bound on distance errors defined as follows: $\forall n, \forall$neighbor n' of n, $|dis_{X\_inf}(n,n') - dis_X(n,n')| \leq e_{rr}$ and $|dis_{Y\_inf}(n,n') - dis_Y(n,n')| \leq e_{rr}$. Physical and logical neighborhood views are defined below; the term "view" is used in this section according to this definition.

Definition 5. *Given a network and a node n that has address ip, we define*

- *physical neighborhood view of* $n \equiv \{\langle ip', dis_{X\_inf}(n,n'), dis_{Y\_inf}(n,n')\rangle \mid ip'$ *is the address of a node* n' *that is physically connected to* n $\}$
- *logical neighborhood view of* $n \equiv \{\langle ip', d_x, d_y\rangle \mid \exists$ *link state packet* $\langle ip, ip', d_x, d_y\rangle\}$

The next theorem states the impact of distance errors on duplication detection. In a network with duplicate addresses, if there exists inconsistency in neighboring addresses, inconsistent views will surely be detected; otherwise by the theorem below, duplication will be detected if any two nodes with the same addresses are not too close.

Theorem 3. *Consider a network in which addresses are unique in two hops neighborhood and nodes that have the same address have the same set of neighboring addresses. At least one node has inconsistent views if any two nodes that have the same address are away at least* $2k \cdot e_{rr}$ *in both x-direction and y-direction, where* e_{rr} *is an upper bound on errors in distance information and k is the length of a special circle defined in the second part of Theorem 2.*

Proof. By Theorem 2, there is a cycle, $\langle ip_0, \ldots, ip_{k-1}, ip_0\rangle$, in the address map such that there exists a circle in the network $\langle$ n_0^0, n_0^1, ..., n_0^{k-1}, n_1^0, n_1^1, ..., n_1^{k-1}, ..., n_{s-1}^0, n_{s-1}^1, ..., n_{s-1}^{k-1}, n_0^0 $\rangle$, where $s \geq 1$ and the address of n_j^i is ip_i

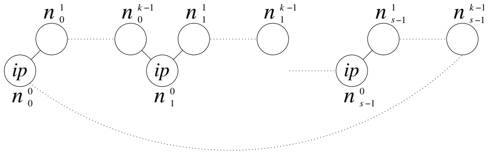


Fig. 5. Proof of Theorem 3

$\forall j \in [0, s-1]$ (Figure 5). We assume in contradiction that all the nodes have consistent views. We define two denotations: (1) the real distance in x-direction from n_i^0 to $n_{(i+1)\%s}^0$: $seg_{Xi} = \sum_{j=0}^{k-2} dis_X\left(n_i^j, n_i^{j+1}\right) + dis_X\left(n_i^{k-1}, n_{(i+1)\%s}^0\right)$; and (2) $seg_{X\_inf} = \sum_{j=0}^{k-2} dis_{X\_inf}\left(n_i^j, n_i^{j+1}\right) + dis_{X\_inf}\left(n_i^{k-1}, n_{(i+1)\%s}^0\right)$. Note the value of seg_{X_inf} does not depend on i, because for all i, $dis_{X\_inf}(n_i^j, n_i^{j+1})$ has the same value since $n_i^j = ip_j$ and $n_i^{j+1} = ip_{j+1}$ and views are consistent on all the nodes.

By the definition of e_{rr}, we have $|seg_{Xi} - seg_{X\_inf}| \leq ke_{rr}$. Since $\sum_{i=0}^{s-1} seg_{Xi} = 0$, we have $seg_{X\_inf} \in [-ke_d, ke_d]$, that is, $seq_{Xi} \in [-2ke_d, 2ke_d]$. So nodes n_i^0 and $n_{(i+1)\%s}^0$ are within $2ke_{rr}$ in x-direction. Similarly, we can prove n_i^0 and $n_{(i+1)\%s}^0$ are within $2ke_{rr}$ in y-direction. So there are two nodes that have the same address and are within distance $2ke_{rr}$ in both x and y direction. Contradiction!

Now we discuss how nodes decide the number of bits to represent distance information if accurate distance information is available. We consider a network in which transmission range of nodes is R. Letting d_x (d_y resp.) be the distance within any two nodes that have the same address in x-direction (y-direction resp.), we have $max\{|d_x|, |d_y|\} \geq \frac{R}{\sqrt{2}}$ by the assumption that addresses are unique within in two hops neighborhood. In order to detect duplication, we require $2ke_{rr} \leq \frac{R}{\sqrt{2}}$, that is, $e_{rr} \leq \frac{R}{2\sqrt{2}k}$. If b bits are used to represent distance information in link state packets, we have $e_{rr} \leq \frac{R}{2^b}$. So all duplications can be detected if $\frac{R}{2^b} \leq \frac{R}{2\sqrt{2}k}$, that is, $b \geq 1.5 \log k$. Nodes can get an upper bound on k by checking lengths of minimal circles in the addresses map; the the "minimal" property of such a circle shown in Theorem 2 implies high possibility of a small k. Note a trivial upper bound on k is the number of *addresses*, which is smaller than the number of nodes or the length of some assumed global unique identification.

6 Conclusion

We investigated duplicate address detection under the assumption that addresses are unique within two hops neighborhood. We propose two definitions of neighborhood views and duplication detection is done by comparing the physical and logical neighborhood views of each node. We show traditional neighborhood information, neighboring addresses, is not sufficient to detect duplicate address, while duplication can be detected by using neighbor distance information that satisfies certain accuracy.

References

1. A.Benlarbi, J. Cousin, R. Ringot, A. Mamouni, and Y. Leroy. Interferometric positioning systems by microwaves. In *Proc. Microwaves Symp.*, Tetuan, Morocco, 2000.
2. P. Boldi and S. Vigna. Universal dynamic synchronous self-stabilization. *Distributed Computing*, 15-3:137–153, 2002.
3. S. Boudjit, A. Laouiti, P. Muhlethaler, and C. Adjih. Duplicate address detection and autoconfiguration in olsr. In *SNPD-SAWN '05*, pages 403–410, Washington, DC, USA, 2005. IEEE Computer Society.
4. J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. In *INFOCOM 2003*, 2003.
5. Y. Chen and E. Fleury. *Duplicate Address Detection in Wireless Ad Hoc Networks Using Wireless Nature*, Research Report, to appear, INRIA, Feb. 2006.
6. B. S. Chlebus, L. Gasieniec, A. Ostlin, and J. M. Robson. Deterministic radio broadcasting. In *Automata, Languages and Programming*, pages 717–728, 2000.
7. A. E. F. Clementi, A. Monti, and R. Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 709–718, Washington, DC, 2001.
8. R. Droms. Dynamic host configuration protocol, 1997.
9. T. K. M. Yamashita. Computing on anonymous networks: Part i — characterizing the solvable cases. *IEEE Transactions on Parallel and Distributed Systems*, 7(1):69–89, 1998.
10. T. K. M. Yamashita. Computing on anonymous networks: Part i — decision and membership problems. *IEEE Transactions on Parallel and Distributed Systems*, 7(1):90–96, 1998.
11. N. Mitton, E. Fleury, I. Gurin-Lassous, B. Sricola, and S. Tixeuil. On fast randomized colorings in sensor networks. technical report LRI-1416, INRIA, Jun. 2005.
12. M. Mohsin and R. Prakash. An ip address configuration algorithm for zeroconf mobile multihop ad hoc networks. In *Int'l. Wksp. Broadband Wireless Ad Hoc Networs and Services*, Sept 2002.
13. S. Nesargi and R.Prakash. Manetconf: Configuration of hosts in a mobile ad hoc network. In *INFOCOM 2002*, June 2002.
14. S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 151–162, Seattle, WA, 1999.
15. R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *IEEE WCNC*, 2002.
16. S. Thomoson and T. Narten. Ipv6 stateless address autoconfiguration. *RFC 2462*, Dec 1998.
17. N. H. Vaidya. Weak duplicate address detection in mobile ad hoc networks. In *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 206–216. ACM Press, 2002.
18. K. Weniger. Passive duplicate address detection in mobile ad hoc networks. In *IEEE WCNC*, New Orleans, LA, Mar 2003.
19. K. Weniger. Pacman: Passive autoconfiguration for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3):507–519, 2005.
20. E. Wesel. *Wireless Multimedia Communications: Networking Video, Voice, and Data.* Addison-Wesley, Reading, MA, 1998.

Fault Monitoring in Ad-Hoc Networks Based on Information Theory

Remi Badonnel, Radu State, and Olivier Festor

MADYNES Research Team,
LORIA-INRIA Lorraine Campus Scientifique - BP 239,
54600 Villers-les-Nancy Cedex, France
{badonnel, state, festor}@loria.fr

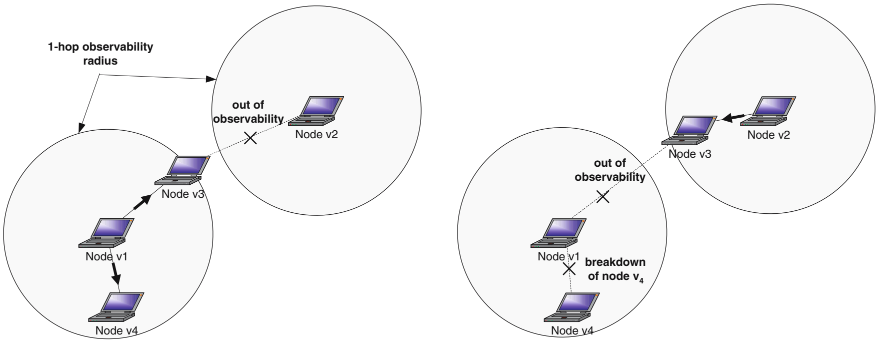
Abstract. Fault detection is a well-known issue in fixed wired networks. Ad-hoc networks provide new challenges towards detecting network failures: the detection task may be hindered by the impossibility to observe a given node. We propose in this paper to monitor the intermittence of network nodes in order to infer network failures. Intermittence can be caused in ad-hoc networks by benign causes due to node mobility and to time-limited out of reachability situations. Abnormal intermittence is however due to faults or malicious network activities. This paper shows how information theoretic measures can identify abnormal intermittence over the routing layer, and proposes a lightweight and distributed intermittence monitoring scheme including several fault detection methods.

Keywords: Ad-Hoc Networks, Network Monitoring, Fault Management.

1 Introduction

Mobile ad-hoc networks [1] are self-configuring networks spontaneously deployed from a set of mobile devices, where a device can interact as a router to forward packets on behalf of the other devices. Our paper addresses the issue of monitoring ad-hoc networks in order to detect faulty behavior. Faulty behavior and intermittence are closely related in ad-hoc networks: a node can have a regular intermittence due to mobility and other ad-hoc specifics, while faulty behavior can generate abnormal intermittence behavior. The key issue that we address in this paper is how to differentiate abnormal intermittence from regular intermittence and thus identify faulty nodes from regular non-faulty ones. While fault detection in fixed wired networks is not hindered by the impossibility to observe a given node, ad-hoc networks specifics do provide major challenges with respect to this issue. A node that does not reply to legitimate polling in an fixed network is typically considered as not functional. In ad-hoc networks, observability is a major issue: a node might not be reachable because it is moving and is out of reachability, or because it is not functioning properly (see figure 1). A centralized manager/agent architecture is not viable for ad-hoc networks, because the manager itself might become isolated or resource might become exhausted. Resource consumption due to management is neglected in fixed networks, while the same

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 427–438, 2006.



(a) Ad-hoc network in phase 1 (b) Ad-hoc network in phase 2

Fig. 1. Fault detection issues in ad-hoc networks. From the perspective of node v_1, both nodes v_3 and v_4 are operational in a first phase. In a second phase, node v_3 goes beyond direct link-level reachability and node v_4 goes down due to faults. From the perspective of node v_1, these two situations are the same.

is of major importance in a landscape where bandwidth and battery lifetime are the key actors. We consider the issue of passive and lightweight monitoring of ad-hoc networks. Monitoring should not generate additional traffic and processing efforts and we thus rely on a passive monitoring approach. We monitor routing level information that is anyway processed by ad-hoc nodes and derive an information theoretic framework [2], where abnormal intermittence can be detected. In order to address the reliability of the monitoring infrastructure, we propose and evaluate several distributed collaborative detection methods. Our approach is centered on a distributed lightweight monitoring scheme, where an entropy derived measure is used to identify abnormal behavior. Our approach is lightweight in the sense that the entropic measure is computed on routing level information which is already available at the node. A distributed and collaborative mechanism is introduced to cope with biased local views.

Our paper is structured as follows : after introducing the monitoring challenges, Section 2 presents our lightweight and distributed monitoring approach for detecting abnormal intermittence of ad-hoc nodes. We briefly overview the routing protocol which serves as an underlying data source in 2.1 and present a failure model for ad-hoc nodes in 2.2. An information theoretic measure for monitoring node intermittence is proposed in 2.3. Several distributed methods of abnormal intermittence detection are described in 2.4 and are evaluated by simulations in Section 3. A survey of related work is given in Section 4. Finally, Section 5 concludes the paper and presents future research efforts.

2 Intermittence Monitoring in Mobile Ad-Hoc Networks

Intermittence in ad-hoc networks is a relative normal condition due to causes that are inherent to such a network: nodes are moving, connectivity might be lost for longer or shorter time-periods and battery life is a well-known issue for this target domain. However, intermittence might have also a different cause related to abnormal ad-hoc behavior, where:

- Failures due to miss-configuration and errors at the physical layer might generate an atypical behavior, where nodes will appear intermittent although from a mobility point of view they did not change significantly,
- Routing failures can be encountered when the routing process is affected by voluntary activity [3], malicious activity (attacks against the routing plane), errors in its configuration or at the protocol stack level,
- Abnormal mobility. While normal mobility is difficult to define, in some specific target deployment (for instance military applications), unpredicted mobility patterns can seriously impact the network resilience and service level.

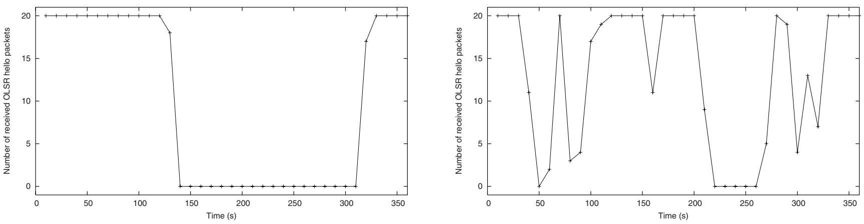
In this paper, we analyze the behavior of intermittent ad-hoc nodes and propose an entropy-based approach for monitoring the routing plane and detecting abnormal intermittent nodes.

2.1 OLSR Routing Protocol Beaconing

The optimized link state routing protocol (OLSR) [4] is a standardized proactive routing protocol that optimizes the pure link state routing algorithms to cope with the requirements of mobile ad-hoc networks. As in a pure link state algorithm, each node determines the list of direct-connected neighbor nodes by accomplishing link sensing through periodic emission of beaconing hello messages. We propose to monitor the routing protocol by analyzing the distribution of hello packets received by each node during the beaconing operation. This is done in order to detect abnormal intermittent ad-hoc nodes. We assume a mobile ad-hoc network as a set of n mobile nodes $V = \{v_1, v_2, ..., v_n\}$ moving in a given surface during a time period T. The time period T is split in k measurement interval $[t_l, t_{l+1}]$ with $t_l = l \times \frac{T}{k}$ for an integer $l \in [0, k]$. During the OLSR beaconing, each node $v_i \in V$ can receive hello packets from the other network nodes located at one hop. The number of beaconing hello packets received by a node v_i from a node v_j is noted X_{v_i,v_j} and can be considered as a random variable $X_{v_i,v_j}(l) : [0, k] \rightarrow [0, b_{max}]$ with l characterizing the interval $[t_l, t_{l+1}]$ and b_{max} the maximal number of hello packets that v_i can receive from v_j. If the hello packets emission interval r is supposed to be homogeneous among network nodes, then X_{v_i,v_j} is bounded by $b_{max} = \frac{1}{r} \times \frac{T}{k}$. This is not a limiting constraint, since the monitoring process can be easily extended to different (per node) r values.

2.2 Ad-Hoc Node Abnormal Intermittence

Statement. Since monitoring is performed at the routing level, we intent to detect abnormal intermittence due to multiple failure causes such as routing



(a) Regular intermittent node

(b) Abnormal intermittent node

Fig. 2. Illustrative examples of the number $X(v_i, v_j)$ of hello packets periodically received by an ad-hoc node

failures, battery problems, physical perturbations and pathological mobility. This monitoring is based on the analysis of how an intermittent node is perceived by a neighbor node or by a set of neighbor nodes. An intuitive idea of intermittence perception by a node can be given by analyzing $X(v_i, v_j)$ values for a network node v_i for different v_j network nodes. Figures 2(a) and 2(b) depict these values respectively for a regular intermittent node and for an abnormal intermittent node. Each figure represents the number of received hello packets $X(v_i, v_j)$ measured (on the y axis) for each time interval $[t_l, t_{l+1}]$ (on the x axis). In figure 2(a), the regular ad-hoc node either generates a short distribution with most of the values equal to 0 when the node is not in the neighborhood, and equals to b_{max} when the node is located in the same neighborhood. In figure 2(b), the abnormal intermittent node (as seen by the other nodes) is characterized by a larger distribution of $X(v_i, v_j)$ values.

Formal model of abnormal intermittence. The main issue that we address is stated in two simple questions. Can we detect abnormal intermittence by monitoring simple parameters like for instance route state related ones? Can we do it in a distributed way such that malicious or non-cooperative nodes are out-weighted? The perception of an abnormal intermittent node by an observing node can be modeled as a discrete Markov chain with four states: these four states depend on the functional state of the observed node (node up or node down), but also on the location of this node compared to the observing node (1-hop neighbor or not). From the perspectives of the abnormal intermittent node itself, the node behavior can be reduced to a discrete Markov chain with two states {NODE UP, NODE DOWN} with the transition probabilities p-failure and q-recovery that a node goes down and respectively goes up after a failure. The stationarity equation can be resolved to get the unique stationary distribution of this irreductible and positive recurrent Markov chain, as presented in equation 1 where p_{up} is the probability to be in state NODE UP and p_{down} is respectively the probability to be in state NODE DOWN.

$$(p_{up}, p_{down}) = (\frac{q}{p+q}, \frac{p}{p+q}) \tag{1}$$

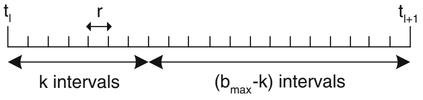


Fig. 3. Measure interval $[t_l, t_{l+1}]$ divided in b_{max} hello emission intervals

In order to evaluate the impact of node abnormal intermittence (parameters p and q) on $X(v_i, v_j)$ distribution, we consider a simple scenario where v_i and v_j are in the same neighborhood, with v_i the observing node and v_j the observed abnormal intermittent node. During the measure interval $[t_l, t_{l+1}]$ (presented in figure 3), the probability of v_j to emit an hello packet at each r hello emission interval is given by p_{up}, the probability that the OLSR node v_j is up. Therefore, the probability for v_i of receiving k hello packets (and then the probability of not receiving $(b_{max} - k)$ hello packets) during $[t_l, t_{l+1}]$ follows a binomial distribution presented in equation 2.

$$P(X_{v_i,v_j} = k) = \sum_{k=0}^{b_{max}} \binom{k}{b_{max}} p_{up}^k (1 - p_{up})^{b_{max}-k} \tag{2}$$

This probability distribution will be considered to determine the impact of transition probabilities p and q on the observed fault behavior.

2.3 Beaconing Entropy Measure

We can monitor the OLSR routing protocol by performing an entropy measure of the probability distribution of $X(v_i, v_j)$. The entropy, defined by Shannon in [5], provides a measure of disorder for a system, where higher values indicate more disordered systems. In our case, it characterizes the distribution disorder (largest distribution) of hello packets for a neighbor node. Equation 3 defines the entropy measure noted $H(X(v_i, v_j))$ in a formal manner.

$$H(X_{v_i,v_j}) = \sum_{k=0}^{b_{max}} P(X_{v_i,v_j} = k).log(\frac{1}{P(X_{v_i,v_j} = k)}) \tag{3}$$

Let us consider the entropy measure for the examples presented in figures 2(a) and 2(b). $H(X_{v_i,v_j})$ equals 1.307 for a regular intermittent node, while $H(X_{v_i,v_j})$ reaches 2.642 for an abnormal intermittent node. High values of $H(X_{v_i,v_j})$ identify a disordered distribution with values $X(v_i, v_j)$ largely covering the interval $[0, b_{max}]$, and thus identify nodes with abnormal intermittence.

Assuming the discrete distribution of $X(v_i, v_j)$ given in equation 2, the entropy $H(X_{v_i,v_j})$ of this binomial distribution can be asymptotically approximated via analytic depoissonization as proposed by Jacquet and Szpankowski in [6] (see equation 4 where a_k are explicitly computable constants).

$$H(X_{v_i,v_j}) \asymp \frac{1}{2} ln(b_{max}) + ln\sqrt{2\pi p_{up}(1-p_{up})} \tag{4}$$
$$+ \sum_{k\geq 1} a_k b_{max}^{-k}$$
$$\asymp ln\sqrt{\frac{2\pi pq}{(p+q)^2}} + c \tag{5}$$

In equation 5, the approximated entropy $H(X_{v_i,v_j})$ is then given in function of the Markov chain's transition probabilities (p, q) (from equation 1) with the constant value $c = \frac{1}{2}ln(b_{max}) + \sum_{k\geq 1} a_k b_{max}^{-k}$. The impact of parameters $(p, q) \in]0, 1[^2$ can be estimated by studying the partial derivatives of $H(X_{v_i,v_j})$. This probabilistic entropy approximation provides an estimate of the additional entropy generated by an abnormal intermittent node (additional to the one generated by the mobility model), perceived from the point of view of a local node. It shows how abnormal intermittent nodes can be detected by a local node, by selecting the network nodes with the highest entropy $H(X_{v_i,v_j})$ of hello packets distribution. The reliability of this local measure can be improved by ad-hoc nodes collaboration.

2.4 Distributed Monitoring Approach

We presented in the previous section how the entropy measure of beaconing packets can locally detect abnormal intermittent nodes. The intermittence detection can be improved by sharing the local measurements among network nodes in a distributed manner. As depicted in figure 4, each ad-hoc node v_1, v_2, v_4, v_5, v_6 monitors locally the network nodes and exchange their local measurements to detect the abnormal intermittent node v_8. We will detail several distributed methods to synthesize the local measurements and to provide a more efficient and reliable intermittence monitoring at the network scale.

A detection approach consists in (1) ranking the potential abnormal intermittent nodes in the ad-hoc network according to a criteria c and then (2) selecting abnormal intermittent nodes according to a threshold value λ (nodes selecting are those presenting a criteria value $c(v_j) > \lambda$). We propose three detection methods and describe them below:

- The first detection method m_1 (called majority voting) defines a ranking of potential abnormal intermittent nodes in function of the number of observing nodes (which perceived the node as abnormal intermittent) in the network.
- The second method m_2 (called entropy sum) takes into account the number of observing nodes, but also the entropy values measured by these nodes. Therefore, m_2 ranks potential abnormal intermittent nodes in function of the sum of entropy values in the network. This method is actually an adaptation of method m_1 where results are weighted by entropy values.
- The last method m_3 (called entropy average) ranks potential abnormal intermittent nodes based on the average of measured entropy values. m_3 does not focus on the number of observing nodes, but favors the entropy values at the network scale.

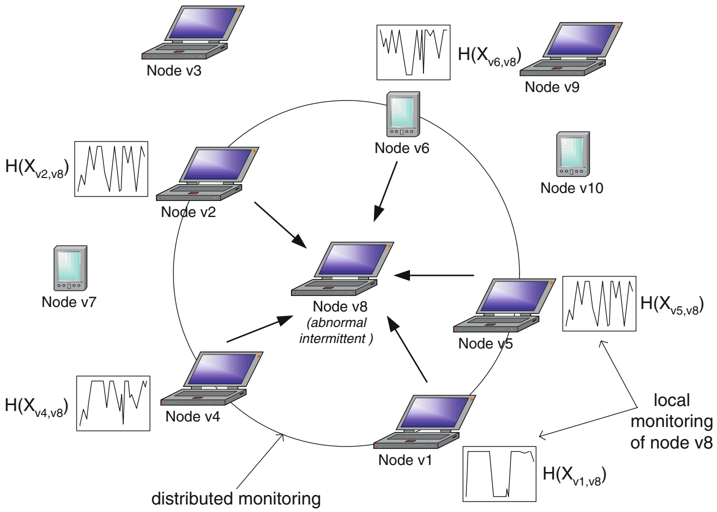


Fig. 4. Distributed intermittence monitoring

These methods can be extended by weighting the measurements obtained from network nodes according to their reliability. Measurements from reliable nodes will have higher weights and then will be more taken into account in the detection process. The temporal coherence and the life time of monitoring data can be improved using approaches such as proposed in [7]. Their performance will be evaluated by simulation in Section 3.

3 Experimental Results

This section describes a set of simulations performed to evaluate the performance of the entropy-based monitoring with the different proposed detection methods, and to estimate the impact of the mobility model on this approach. The experiments were performed with the discrete event network simulator ns-2 [8]. We simulated a mobile ad-hoc network of 50 nodes moving in a 1500 m x 300 m rectangular area during a time period of 900 simulated seconds. To avoid initialization discrepancy issues with the mobility model [9], we used the steady-state mobility model generator *mobgen-ss* where initial speeds and locations of nodes are chosen from the stationary distribution to perform an immediate convergence and provide more reliable simulations. For each experiment, a set of abnormal intermittent nodes is randomly chosen and follows the two-state Markov chain model with transition probabilities (p, q). This set of abnormal intermittent nodes is then compared to the set of nodes detected as abnormal intermittent nodes by the detection scheme.

In order to quantify the performance of the approach, we performed an analysis of sensitivity and specificity. Our approach can be seen as a diagnostic test, where we test if an ad-hoc node is abnormal intermittent (positive test) or

Table 1. Simulation parameters

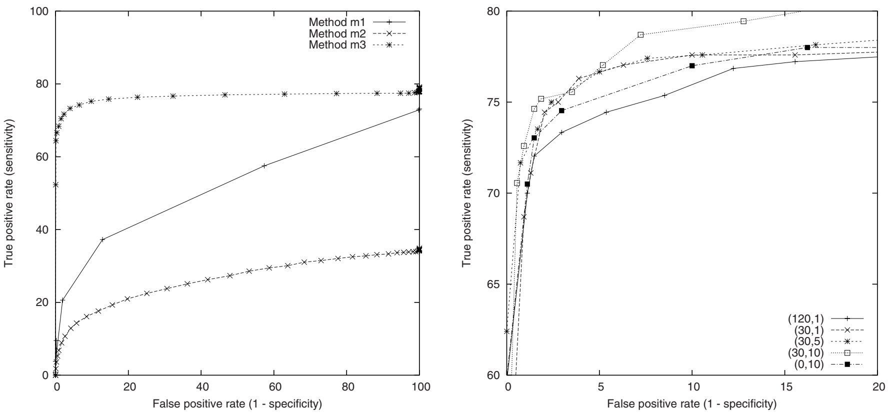
Parameter	Value
Simulator	ns-2
Simulation time	900 s
Simulation area	1500 m x 300 m
Number of ad-hoc nodes	50 nodes
Number of abnormal nodes	0 - 5 node(s)
Mobility model	random waypoint *mobgen - steady state*
Speed	0.1 - 10 m/s
Pause time	0 - 120 s
Physical Layer	FSP / 2-RGR
MAC layer	IEEE 802.11
Routing layer	NRL OLSR

regular intermittent (negative test). The sensitivity shows how well the method picks up true cases (true positive or true negative results), while the specificity defines how well it detects false cases (false positive or false negative results). We use the receiver operating characteristic (ROC) [10], a graphical plot of sensitivity (Sn) versus 1-specificity (1 - Sp), to evaluate the detection efficiency. The ideal diagnostic method shows a plot that is a point in the upper left corner of the ROC space, as sensitivity (all true positives are found) and specificity (no false positives are found) reach both 1.0. A diagnostic method becomes random (and then inefficient) when it presents a line at an 45 degree angle from bottom left to top right, because the number of true positives equals the number of false positives. In the next parts of this section, we will detail the experimental results (1) by plotting and analyzing the ROC curves to compare the performance of the three detection methods and (2) by evaluating the impact of mobility model (random waypoint model with parameters $(pause, speed)$).

3.1 Performance of the Collaborative Detection Methods

In a first set of experiments, we analyzed the performance of the three collaborative detection methods. These results are shown in figure 5(a) and are based on an extensive set of simulations with different mobility parameters $(pause, speed)$ and abnormal intermittence parameters (p, q). We varied node mobility with pause time *pause* from 0 to 120 s and with speed *speed* from 0.1 to 10 m/s. The abnormal intermittent nodes were parameterized with realistic transition probabilities. The failure probability p was set with low values from 0.1 to 0.2 and the recovery probability q from 0.1 to 1.0. For each individual setting we performed 150 simulations to assure the non-bias of the result.

The performance of the detection methods is summarized on figure 5(a), where we plotted the ROC curve for each method. A point (x,y) on a curve stands for the true positive rate (y) of the method compared to the false positive rate (x) for a given threshold value. We are interested in an optimal diagnostic method providing a low false positive rate for a maximum true positive rate. The closer a method is localized in the upper left corner of the ROC space, the more it provides an efficient detection. We can therefore deduct that method m_3 based



(a) Comparison of methods m_1, m_2, m_3 (b) Impact of mobility on method m_3

Fig. 5. Evaluation of the collaborative detection methods with ROC curves

on the average of entropies presents a better diagnostic test than the two others. In particular, method m_3 offers good results with a true positive rate of more than 70% in most of cases. In a more refined way, if we expect a false positive rate of less than 20%, method m_3 with 70% of true positive is definitively better than method m_1 providing a true positive rate of less than 45%, and still better than method m_2 showing a true positive rate of less than 20%. It turns out that methods m_1 and m_2 present less convincing performance, which can come from the simple fact that the detection is too dependent on the number of nodes observing an intermittent node. For instance for method m_1, the detection is based on the majority voting and consequently the probability of an ad-hoc node to be detected grows with the neighbor number of that node. In the same way, method m_2 considers the entropy sum at the network scale, which also raises in function of the number of neighbor nodes. In method m_3, using the average of entropies provides a more independent and reliable measurement of intermittence, where the increasing of the number of neighbors improves and refines the averaged measurement without denaturing it.

3.2 Impact of Mobility Model on Intermittence Detection

A natural question is whether mobility impacts the performance of the abnormal intermittence detection. Intuitively, higher mobility should make things worse: the entropy generated by mobility should hide the entropy generated by abnormal intermittence, but a precise quantification of this effect is required. A second series of experiments addressed this issue, where different mobility parameters were evaluated with a realistic intermittence (parameters $p = 0.1$ and $q = 0.4$). We varied the random waypoint parameters with reasonable pause time from 0 to 120 s and speed from 1 to 10 m/s, and measured the sensibility and specificity of the entropy average method m_3. These results are presented in figure 5(b) where we plotted the ROC curves for each couple $(pause, speed)$ of mobility parameters. We were interested in studying the detection method for configurations with low false positive rate and we therefore limited the plotting of ROC curves to a false positive rate no more than 20%. The comparison of ROC curves shows that the impact of mobility is relatively limited for realistic mobility scenarios. The variation between the lowest and the highest mobility parameters is indeed less than 5%. This statement comes from the nature of our measure, which actually highlights more the additional entropy generated by abnormal intermittence than the entropy generated by the network mobility. We expected that higher mobility implies bad results (i.e.: high speeds and short pause times), where by a low result we understand a low true positive rate for a false positive rate of less than 20 %. Such was the case indeed (note the case of $pause = 0$ and $speed = 50$) where the sensibility is less than 72%. A rather surprising result is however the case of lower mobility parameters where the sensibility is improved when mobility grows. This contradicts our initial hypothesis that mobility deteriorates our detection and leads us to more contrasted conclusion. The sensibility actually evolves in two steps. First, the detection is improved by mobility rise, from mobility parameters $(120, 1)$ to $(30, 10)$. The mobility increases the number of observing nodes per observed node. Second, the detection is noised with highest mobility scenarios and is not capable anymore to highlight efficiently abnormal intermittent nodes. In brief, the detection shows best results when mobility scenarios are not extreme (lowest and highest mobility parameters).

4 Related Work

Among the pioneering approaches in our context of fault management (we do not focus on intrusion detection/security), Jakobson introduces in [11] an approach for correlating events and faults with temporal constraints. Failures detection algorithms based on keep-alive messages (active approach) are experimented in [12] and their performance are evaluated in overlay networks. The OLSR hello mechanism corresponds to one of the experimented keep-alive approaches called gossip approach where a node periodically sends "I'm alive" messages to its neighbors. Related work in monitoring the routing plane for fixed networks is described in [13], where a real-time system tracks the routing state of a single OSPF domain, using flexibly OSPF snooping and link state SNMP tracking. This system offers network statistics based on the monitoring of a link-state routing protocol, but it is mainly designed for performance analysis rather than fault detection. The DAMON architecture [14] defines a distributed monitoring system based on agents for multi-hop networks: agents perform the network monitoring and send to data repositories the measurements data. DAMON supports multiple data repositories and includes an auto-discovery mechanism of data repositories by the agents. This generic architecture is not dedicated to specific network parameters and could therefore be appropriate for the storage of fault monitoring data. WANMon is a monitoring tool described in [15] to monitor the resource usage in terms of network traffic, energy, memory and CPU, but its scope is limited to the host-level monitoring. Finally, our previous work in [16, 17] addresses an information model and a probe-based architecture for monitoring ad-hoc node participation.

5 Conclusions

We proposed in this paper a lightweight and distributed fault monitoring approach for ad-hoc networks and addressed the issue of detecting abnormal intermittence of ad-hoc nodes. The proposed solution is based on two key concepts: (1) a measure based on information theory to monitor intermittence over the routing layer and (2) a distributed scheme to perform abnormal intermittence detection among the network nodes. We have shown how correlating monitored data from different ad-hoc hosts provides an efficient and reliable detection of abnormal intermittence. We have proposed and evaluated different distributed methods based on fault ranking and thresholding methods. The main advantages of our approach are multiple: we can detect abnormal intermittent nodes even if they are not instrumented. The monitoring process is passive and completely decoupled from the OLSR protocol, without requiring any additional routing protocol piggybacking. Our future work will consist in assessing the intermittence monitoring with respect to different mobility models, defining an autoconfiguration mechanism for the collaborative detection methods and integrating the monitoring scheme into a management architecture.

References

1. Basagni, S., Conti, M., Giordano, S., Stojmenovic, I., eds.: Mobile Ad Hoc Networking. IEEE Press and John Wiley & Sons, Inc., Piscataway, NJ and New York, NY (2004)
2. Cover, T., Thomas, J.: Elements of Information Theory. Wiley & Sons (1991)
3. Kherani, A., Altman, E., Michiardi, P., Molva, R.: Non-cooperative Forwarding in Ad-hoc Networks. In: Proc. of the International IFIP Networking Conference (Networking'05), Waterloo, Canada (2005)
4. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). http://www.ietf.org/rfc/rfc3626.txt (2003) IETF RFC 3626.
5. Shannon, C.E.: A Mathematical Theory of Communication. The Bell System Technical Journal **27** (1948) 379–423
6. Jacquet, P., Szpankowski, W.: Entropy Calculation via Analytic Depoissonization. IEEE Transaction on Information Theory **45** (1999) 1072–1081
7. Westphal, C.: On Maximizing the Lifetime of Distributed Information in Ad-Hoc Networks with Individual Constraints. In: Proc. of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'05), Urbana-Champaign, IL, USA (2005)
8. SAMAN: NS-2 Network Simulator. http://www.isi.edu/nsnam/ns/ (1989)
9. Yoon, J., Liu, M., Noble, B.: Random Waypoint Considered Harmful. In: Proc. of IEEE International Conference on Computer Communications (INFOCOM'03), San Francisco, CA, USA (2003) 1312–1321

10. Zweig, M., Campbell, G.: Receiver-Operating Characteristic (ROC) Plots: a Fundamental Evaluation Tool. Clinical Chemistry **29**(4) (1993) 561–577
11. Jakobson, G., Weissman, M.D.: Real-time Telecommunication Network Management: Extending Event Correlation with Temporal Constraints. In: Proc. of the 4th IFIP/IEEE International Symposium on Integrated Network Management (IM'95), Santa Barbara, CA, USA (1995)
12. Zhuang, S.Q., Geels, D., Stoica, I., Katz, R.H.: On Failure Detection Algorithms in Overlay Networks. In: Proc. of IEEE International Conference on Computer Communications (INFOCOM'05), Miami, FL, USA (2005)
13. Baccelli, E., Rajan, R.: Real-Time OSPF Route Monitoring. In: Proc. of the 7th IFIP/IEEE International Symposium on Integrated Network Management (IM'01), Seattle, WA, USA (2001)
14. Ramachandran, K., Belding-Royer, E., Almeroth, K.: DAMON: A Distributed Architecture for Monitoring Multi-hop Mobile Networks. In: Proc. of IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'04), Santa Clara, CA, USA (2004)
15. Ngo, D., Wu, J.: WANMON: a Resource Usage Monitoring Tool for Ad-hoc Wireless Networks. In: Proc. of the 28th Annual IEEE Conference on Local Computer Networks (LCN'03), Bonn, Germany (2003) 738–745
16. Badonnel, R., State, R., Festor, O.: Management of Mobile Ad-Hoc Networks : Evaluating the Network Behavior. In: Proc. of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM'05), Nice, France (2005) 17–30
17. Badonnel, R., State, R., Festor, O.: Management of Mobile Ad-Hoc Networks: Information Model and Probe-based Architecture. ACM International Journal of Network Management (ACM IJNM) **15**(5) (2005)

Performance Analysis of Exposed Terminal Effect in IEEE 802.11 Ad Hoc Networks in Finite Load Conditions

Dimitris Vassis and Georgios Kormentzas

Dept. of Information and Communication Systems Engineering,
University of the Aegean GR-83200, Karlovassi, Greece
{Divas, gkorm}@aegean.gr

Abstract. The paper evaluates the performance effects of exposed terminals in IEEE 802.11 ad hoc networks in finite load conditions. In this context, employing also models from previous authors' work, the paper derives analytical models for the estimation of channel utilization and media access delay for IEEE 802.11 ad hoc networks in finite load conditions with and without exposed terminals. The simulation results show that the analytical estimated channel utilization and media access delay metrics are fairly accurate.

1 Introduction

Due to the lack of a centralized control entity in ad hoc networks, sharing of wireless bandwidth among ad hoc terminals has to be organised in a decentralised manner. In this context, distributed Medium Access Control (MAC) mechanisms such as the IEEE 802.11 Distributed Coordination Function (DCF) [1], have gained widespread popularity in ad hoc networks. As all Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) based MAC protocols, DCF suffers from hidden and exposed terminal problems. The use of Request to Send/Clear to Send (RTS/CTS)-like schemes partially solves the hidden terminal problem, while it leaves the exposed terminal open, where some nodes that heard the RTS/CTS exchange refrain from transmission even though they would not have interfered with any going transmission, thus the result for the whole system is channel utilization and throughput reduction.

As the exposed terminal problem results in performance degradation of IEEE 802.11 ad hoc networks and DCF does not implement an appropriate mechanism to alleviate this problem, there are a lot of works at the international literature where the authors propose their own solutions by enhancing DCF (e.g., [2-4]). Independently of the various proposed solutions, the paper discusses the performance effects of exposed terminals in IEEE 802.11 ad hoc networks in finite load conditions. In this context, the paper derives appropriate fairly accurate analytical models for the estimation of channel utilization and media access delay for IEEE 802.11 ad hoc networks in finite load conditions with and without exposed terminals.

The rest of the paper is organised as follows. Section 2 gives a brief description of the DCF mechanism and presents the hidden and exposed terminal problems. Section 3 and 4 discus the proposed analytical utilization and delay models respectively, and Section 5 validates the accuracy of the analytical estimated metrics through appropriate simulation scenarios. Section 6 evaluates the performance effects of exposed

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 439 – 450, 2006.

terminals in IEEE 802.11 ad hoc networks and finally, Section 7, gives the concluding remarks of this work.

2 Brief Description of the IEEE 802.11 DCF Mechanism

According to DCF, a node can initiate a transmission only if it senses the medium as being idle for a time interval greater than a Distributed InterFrame Space (DIFS). If a collision occurs, the transmission is deferred and a backoff process starts. Unlike wired networks (with Carrier Sense Multiple Access with Collision Detection support), in a wireless environment, collision detection is not possible. Hence, an acknowledgement (ACK) frame is used to notify the sending node that the transmitted data has been successfully received. The transmission of ACK is initiated at a time interval equal to Short InterFrame Space (SIFS) that follows the reception of the sending data (see Figure 1(a)).

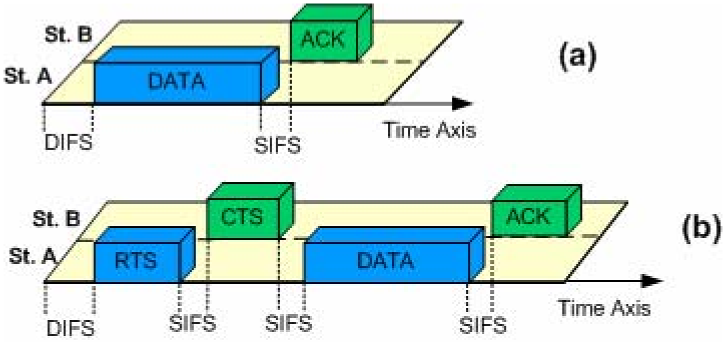
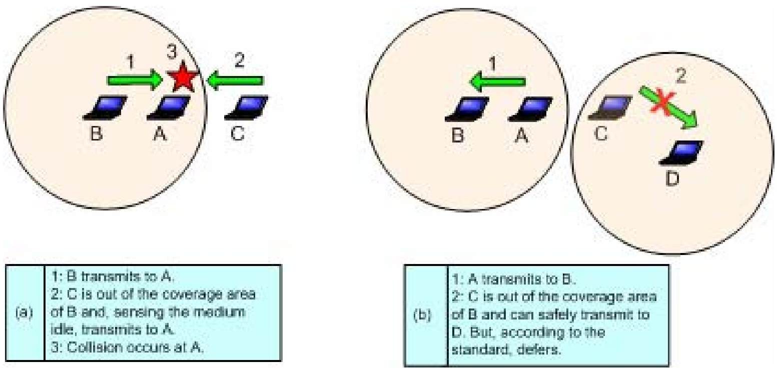


Fig. 1. Frame exchanges in a packet transmission procedure

The above transmission mechanism does not protect the wireless nodes from the hidden terminal problem (see Figure 2(a)). This problem arises, for example, when for



(a) 1: B transmits to A.
2: C is out of the coverage area of B and, sensing the medium idle, transmits to A.
3: Collision occurs at A.

(b) 1: A transmits to B.
2: C is out of the coverage area of B and can safely transmit to D. But, according to the standard, defers.

Fig. 2. The hidden and exposed terminal problems

three stations, namely A, B and C, A is able to communicate with B and C, but C is not within the range of B. In this case, C is hidden for B. This means that, when B transmits a frame to A, C may sense the medium as being idle. If C also transmits a frame towards A, then a collision will occur at A.

In order to alleviate this problem, the IEEE 802.11 standard includes a virtual carrier sense mechanism, which is based on the exchange of two short control frames: a Request To Send frame (RTS), which is sent by a potential transmitter to the receiver and a Clear To Send frame (CTS), which is sent back from the receiver in response to RTS. The RTS and CTS frames include a duration field that specifies for a station the time interval necessary to completely transmit its data and the related ACK. Other stations can hear either the sender or the receiver and refrain from transmitting until the data transmission is complete (see Figure 1(b)). The RTS/CTS mechanism adds a considerable overhead in the medium, especially for the transmission of small data packets. In this context, the use of RTS/CTS is controlled through the RTS Threshold attribute, where only packets with size greater than the value of the RTS Threshold are transmitted with the RTS/CTS mechanism.

The exposed terminal problem is reverse to the hidden terminal one. Taking as reference Figure 2(b), A is able to communicate with B and C. C is able to communicate with A and D but it is outside the coverage area of B. Now, during a frame transmission from A to B, C senses the medium busy and defers from transmitting a frame to D, despite the fact that this transmission would not result in a collision, as C is outside the coverage area of B, and A is outside the coverage area of D. In this case C is exposed to A. Contrary to the hidden terminal problem, as already mentioned in the introductory section, the IEEE 802.11 standard does not include an appropriate mechanism to alleviate the exposed terminal problem.

3 The Proposed Analytical Utilization Model

This section presents an analytical model that estimates the channel utilization effect of exposed terminals in IEEE 802.11 ad hoc networks. In order to derive this estimation, the paper employs previous authors' work [5] concerning the analytical estimation of channel utilization for IEEE 802.11 ad hoc networks under the hidden terminal problem. Note that the channel utilization is considered to be the percentage of time in which useful information is transmitted in the wireless medium. Firstly, we give a brief overview of the model presented in [5] and then we proceed to estimate the channel utilization effect of the exposed terminals.

Figure 3 depicts a conventional IEEE 802.11 ad hoc network where the nodes' transmission range is r and the wireless channel data rate is R. Let now consider the node A of Figure 3. The small circle includes the nodes that are inside the coverage area of node A. The big circle contains also the hidden nodes for node A, which are the nodes at the ring between the big and the small circles.

Our work in [5] derives the channel utilization in respect to the aggregate traffic produced in the small circle. [5] assumes that the aggregate offered traffic load in the

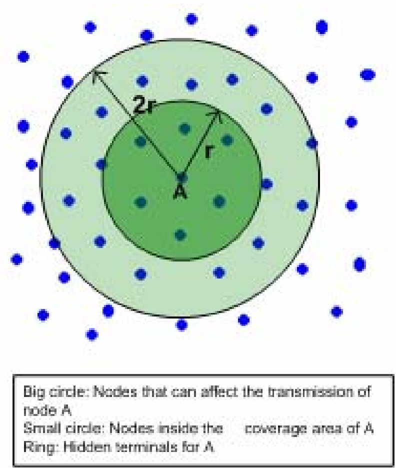

2r
r
A
Big circle: Nodes that can affect the transmission of node A
Small circle: Nodes inside the coverage area of A
Ring: Hidden terminals for A

Fig. 3. A conventional IEEE 802.11 ad hoc network

wireless channel is generally distributed with a mean value of g packets per second and that the payload size of packets transmitted in the wireless medium is also generally distributed with a mean value of P bits. It also assumes that the network operates under finite load conditions, so, the offered traffic consists not only of new packets but also of previously collided packets.

According to [5], the channel utilization related to a random node of a conventional IEEE 802.11 ad hoc network is

$$U = \frac{S}{(B+I)^4} I^3 p^{3T_s/\tau} \tag{1}$$

where S is the average transmission time of the packet payload in the wireless medium, B is the average busy period of the medium, I is on average the idle period of the medium, τ is the duration of an IEEE 802.11 time slot [1], p is the probability that no packet arrives in a time slot, and T_s is the time needed for a successful transmission.

Again according to [5],

$$I = \frac{\tau}{1-p}, \tag{2}$$

$$B = \frac{T_c + p_s(T_s - T_c)}{p}, \tag{3}$$

$$S = \frac{Pp_1}{Rp(1-p)}, \tag{4}$$

$$p_s = \frac{p_1}{1-p}, \tag{5}$$

where T_c is the time a collision takes (explicit expressions for T_s and T_c are given in [6]), and p_1 is the probability that a single arrival occurs in a time slot.

3.1 Analysis of the Utilization Effect of the Exposed Terminals

Let now take a look at Figure 4, where we consider that a transmission is in progress from node A to a random node inside the coverage area of A, i.e., a circle with centre A, named *E*. We furthermore assume that this transmission does not refer to node B. According to the IEEE 802.11 standard, if node B has a packet to send, it has to wait until the transmission concerning A is completed.

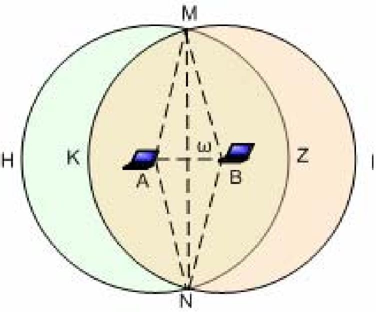


Fig. 4. Exposed terminal effect analysis

If the transmitted packet from A is destined to a node inside the area E_1 that is the area (NKMZ), then node B has to defer its transmission, as otherwise a collision will occur to the receiving node. However, if the transmitted packet from A is destined to a node inside the area E_2 that is the area (NHMK), then node B could not have to defer as the receiving node is outside its coverage area. But, still, in this second case, according to the standard, B has to defer. Hence, by assuming that nodes are uniformly distributed across the network area and transmissions occur equiprobably to all nodes, when node B wants to transmit and the medium is busy, it has to wait pointlessly in a fraction E_2 / E of its anticipated transmissions.

We proceed now to examine the effect of the above pointless deferrals in the network's utilization. Assume that the IEEE 802.11 standard includes a sophisticated solution in the exposed terminal problem, such that a node knows when to defer from transmission while another one transmits, and when not. The probability that a deferral occurs because the medium is busy is equal to the probability that the medium is busy, which is $B/(B+I)$, times the probability that a transmission occurs in a slot, which is $(1-p)$. If nodes know when to transmit during another transmission and when not, then a percentage $\beta = \mathrm{B}/(\mathrm{B}+\mathrm{I})\cdot(1-p)\cdot(E_2/E)$ of additional transmissions will occur in the medium, simultaneously with transmissions that are already in progress. This means that we will have a virtual additional channel where transmissions occur with packet generation rate equal to g_e, such that the probability that a packet arrives in a time slot is β. Apparently, the parameter g_e can be easily derived from β if we know the distribution describing the packet generation rate.

The fraction E_2/E is derived in Appendix equal to 0.42. Accordingly, the value of β from which g_e arises is

$$\beta = \frac{0.42\mathrm{B}(1-p)}{B+I}. \tag{6}$$

Eventually, the extra channel utilization if the exposed terminal problem would not exist can be derived from Equation 1 by replacing g with g_e. Accordingly, the total channel utilization in the ideal absence of the exposed terminal problem is

$$U = U(g) + U(g_e). \tag{7}$$

4 The Proposed Analytical Delay Model

This section discusses an analytical model that estimates the media access delay effect of exposed terminals in IEEE 802.11 ad hoc networks in finite load conditions. The media access delay is the time from the beginning of a packet transmission from a node till its successful reception from the next node.

For the media access delay estimation, consider again a node (e.g., node A of Figure 2) and a cell that includes all nodes in the coverage area of the considered node. Let G be the normalized traffic of a cell with respect to the packet transmission time, given as $G = g\cdot P/R$. Moreover, the mean number of retransmissions before a packet is sent is G/U [7]. Excluding the last, successful transmission, the actual mean number of retransmissions is $m = (G/U)-1$. Considering a large packet inter arrival time compared to the time slot, we can assume that the packet does not wait for a DIFS interval in the first transmission attempt. The expected media access delay of a packet transmitted from a node is equal to the duration T_s needed for the packet to be successfully transmitted plus the time taken for all its unsuccessful transmissions. The second is equal to the expected duration of each unsuccessful transmission (backoff time plus the duration T_c of a collision) times the number of unsuccessful transmissions, m. Accordingly, the media access delay is given as

$$d_m = T_s + m(T_c + \tau\overline{X} + \overline{F}), \tag{8}$$

where $\overline{X}$ is the expected number of backoff slots in a retransmission and $\overline{F}$ is the time where the backoff counter freezes because of transmissions that are in progress.

Considering that the backoff slots in each attempt are uniformly distributed between 1 and the contention window, then

$$\overline{X} = \frac{1}{m}\sum_{j=0}^{m-1} W2^{j} = \frac{1}{2m} W2^{m-1}, \tag{9}$$

where W is the minimum contention window.

For the time $\overline{F}$ that the backoff counter freezes because of another transmission, consider that the probability that the backoff counter expires is simply $1/\overline{X}$. Under the finite load conditions of our case, the transmitter of a node can be thought as a G/G/1 queue with utilization factor $u = g_N d_m$, where d_m is the media access delay and g_N is the node traffic. The probability that a node transmits is equal to the probability that the backoff counter expires, given that it has a packet in the transmitter. This is $u/\overline{X}$. Hence, if M is the number of nodes in a cell, in $\overline{X}$ slots where the backoff counter decreases, there will occur $M \cdot (U/\overline{X}) \cdot \overline{X} = M \cdot g_N \cdot d_m =$ $g \cdot d_m$ additional transmissions. From all these transmissions there are $g \cdot d_m$ $[m/(m+1)]$ retransmission and $g \cdot d_m[1/(m+1)]$ successful transmissions. In this way all these transmissions add an additional backoff time of

$$\overline{F} = g \cdot d_m [T_s/(m+1) + mT_c(m+1)]. \tag{10}$$

Eventually, the media access delay can be derived by substituting $\overline{F}$ into Equation 8, giving

$$d_m = \frac{m(T_c + \tau\overline{X}) + T_s}{1 - gm(Ts + mT_c)/(m+1)} \tag{11}$$

4.1 Analysis of the Delay Effect of the Exposed Terminals

For deriving the media access delay if the exposed terminal problem would not exist, we assume again that the IEEE 802. 11 standard includes a sophisticated algorithm where nodes know when to transmit and when not, during another transmission in progress. Following this line of thought, if a node freezes its backoff counter for $g \cdot d_m$ transmissions that occur from other nodes (as denoted above), then in the case where the exposed terminal problem is solved, it would only freeze the backoff counter for a fraction $(1 - E_2/E)$ (that is 0.58) of these transmissions only. Consequently, Equation 10 is rearranged as follows:

$$d_m = \frac{m(T_c + \tau\overline{X}) + T_s}{1 - 0.58gm(Ts + mT_c)/(m+1)} \tag{12}$$

In the above Equation, m is derived through U from Equation 7.

5 Models Validation

In order to validate the accuracy of the derived in the previous sections analytic approximations of channel utilization and media access delay effects of exposed terminals in ad hoc networks performance, several simulation scenarios were considered in the Pythagor simulation platform [8], an open C++ simulation tool for IEEE 802.11 a/b/g networks.

The estimated models stand for any input traffic model, as long as the parameters g , p , p_1 and P are given in a closed form. For our validation tests, although the input traffic depends on many characteristics specific to each network, the Poisson distribution with a mean value of g packets/s was adopted to efficiently characterize the aggregated generated traffic inside a cell. Consequently, p , p_1, and g_e (for the exposed terminal problem analysis) can be easily derived as follows:

$$\begin{aligned} p &= \frac{(g\tau)^0}{0!} e^{-g\tau} = e^{-g\tau} \\ p_1 &= \frac{(g\tau)^0}{0!} e^{-g\tau} = g\tau e^{-g\tau} \\ \beta &= 1 - e^{-g_e\tau} \Rightarrow g_e = -\ln(1-\beta)/\tau \end{aligned} \quad . \qquad (13)$$

In addition, an exponential distribution with a mean value of P bits was chosen for the packet payload size. The exponential distribution is proven adequate to describe the packet size distribution in IEEE 802.11 networks (e.g., [6]).

Table 1 summarizes the input parameters used during the validation process.

Table 1. Simulation parameters

Parameter	Value
time slot duration (τ)	20μs
SIFS	10μsec
DIFS	50μsec
minimum contention window (W)	31slots
PHY header	96bits
MAC header	272bits
ACK	112bits + PHY
RTS	160bits + PHY
CTS	112bits + PHY
Channel data rate (R)	5.5Mb/s
RTS Threshold	128Bytes
packet payload (P)	1KByte

Tables 2 and 3 summarize the results of the comparison between the analytical metrics and the Pythagor output. In all cases the results show that the models are fairly accurate.

Table 2. Model validation for the channel utilization effect of exposed terminals

Channel traffic (g) Kb/s (1 packet = 8*P* bits)	**Channel utilization (%) (Equation 7)**	**Channel utilization (%) (Pythagor)**	**Difference (%)**
1000	18.94	20.45	8
2000	18.16	19.61	8
3000	16.88	18.07	7
4000	15.32	16.39	7

Table 3. Model validation for the media access delay effect of exposed terminals

Channel traffic (g) Kb/s (1 packet = 8*P* bits)	**Media access delay (ms) (Equation 12)**	**Media access delay (ms) (Pythagor)**	**Difference (%)**
1000	2.42	2.63	9
2000	5.23	5.65	8
3000	9.78	10.56	8
4000	562.31	601.68	7

6 Performance Evaluation of Exposed Terminal Effect

This section evaluates the utilization and delay effects of exposed terminals in an IEEE 802.11 ad hoc network that employs the parameters of Table 1. We concern three cases, where in the first case, we assume clear channel conditions, where all nodes are in line of sight with each other, in the second one, we assume real conditions, meaning existence of hidden and exposed terminals, and in the third one, we assume that the standard includes a sophisticated solution such that exposed terminals do not exist.

As it is shown in Figure 5 and obviously expected, the channel utilization for clear channel conditions is remarkably higher than the other two cases. Another important difference concerns the shapes of the three curves. This difference graphically depicts the effect of the exposed terminals. In the clear channel conditions case where both the hidden and the exposed terminal problems do not exist, most collisions are prevented through the RTS/CTS mechanism. As a result, even if the input traffic increases too much, the utilization does not decrease. In the real conditions case where both problems exist, hidden nodes transmit during the transmission of others and exposed nodes defer their transmissions pointlessly. Consequently, as the network traffic increases, so do the collisions caused from the aforementioned problems, resulting in rapid utilization impairment, which is depicted through the sharp shape of

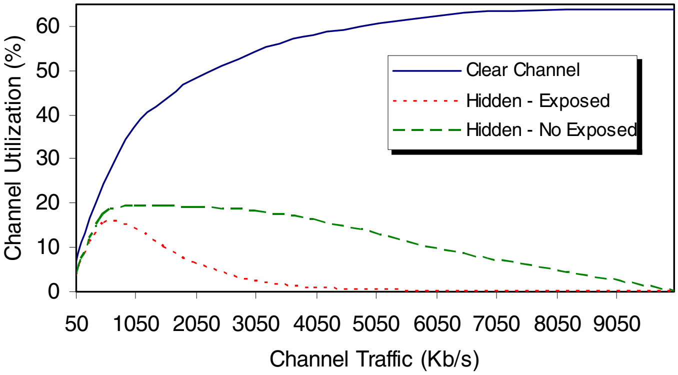


Fig. 5. Channel utilization effect of exposed terminals

the curve. In the case where the exposed terminal problem is treated, the collisions are less and the utilization impairment is slower and consequently the shape of the respective curve is smoother.

Figure 6 depicts the media access delay effect, which is totally aligned with the utilization one. The media access delay for clear channel conditions is very small. Moreover, it does not increase rapidly as in the other two cases. This is because the utilization does not decrease after a limit of channel input traffic, so the number of retransmission attempts increases at a rate less than linear. Furthermore, as it is expected the absence of exposed terminals keeps the media access delay remarkably lower than happens in real conditions case.

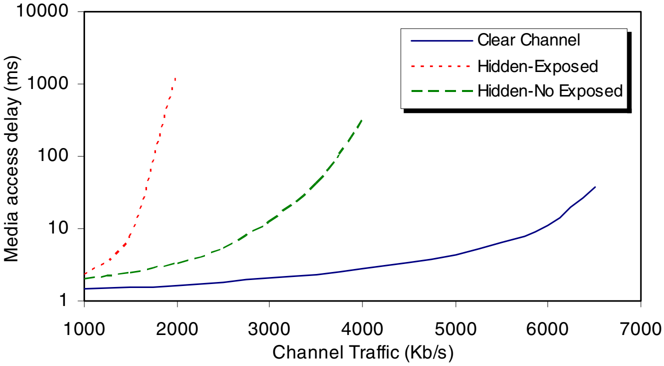


Fig. 6. Media access delay effect of exposed terminals

7 Conclusions

The paper presents analytical models for the estimation of channel utilization and media access delay effects of exposed terminals in IEEE 802.11 ad hoc networks in finite load conditions. In order to validate the accuracy of the analytic approximations, several simulation scenarios were considered in the Pythagor simulation platform. In all cases the results of the comparison between the analytical metrics estimate and the Pythagor output show that the models are fairly accurate.

Acknowledgment

The work reported in this paper is supported in part by 'Pythagoras II – Research Group Support of the University of the Aegean'.

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1999.

[2] F. Borgonovo, A. Capone, M. Cesana, L. Fratta, "ADHOC MAC: a new, flexible and reliable MAC Architecture for ad hoc Networks", *In Proc. of 2003 IEEE Wireless Communications and Networking Conference*, 16-20 March 2003, New Orleans LA, USA, Volume: 2, Page(s): 965-970.

[3] D. Shukla, L. Chandran-Wadia and S. Iyer, "Mitigating the Exposed Node Problem in IEEE 802.11 Ad Hoc Networks", *In Proc. IEEE CCCN03*, October 2003, USA.

[4] A. Acharya, A. Mishra, and S. Bansal, "MACA-P: A MAC for Concurrent Transmissions in Multi-hop Wireless Networks", *IBM Research Report RC22528*, July 2002, *In Proc. IEEE PerCom*, 2003.

[5] D. Vassis and G. Kormentzas, ""Throughput Analysis for IEEE 802.11 Ad Hoc Networks under the Hidden Terminal Problem"", In Proc. *IEEE CCNC2006 HWN-RMQ Workshop*, Nevada, USA, January 2006.

[6] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas of Communications, vol. 18, no. 3*, pp. 535-547, 2000.

[7] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I-Carrier Sense Multiple Access Models and their Throughput-Delay Characteristics," *IEEE Transactions on Communications, vol. 23, no. 12*, pp. 1400-1416, Dec. 1975.

[8] Pythagorsimulation tool, Available on line at the URL: http://www.icsd.aegean.gr/ telecom/pythagor/index.htm

Appendix: Deriving the fraction E2/E

As the fraction E_2 / E depends on the distance between the transmitting node (A) and the exposed node (B), we will derive it for the average distance between two nodes inside a cell. For two nodes A and B, the probability that B is x meters away from A is $2\pi x dx / \pi r^2$. Consequently, the mean distance d is

$$\bar{d} = \int_0^r \frac{2\pi x}{\pi r^2} dx = \frac{2}{3} r$$

Concerning Figure 4 and the notation used in Section 3, we need to derive the fraction $(MHNK)/\pi r^2$. For $\mathrm{d} = (\mathrm{AB}) = 2\mathrm{r}/3$, we have:

$$
\begin{aligned}
&(\mathrm{MHNK}) = \pi\pi^2 \text{ - } (\mathrm{MKNZ}),\\
&(\mathrm{MKNZ}) = 2(\mathrm{MKN}),\\
&(\mathrm{MKN}) = (\mathrm{BMKN}) \text{ - } (\mathrm{BMN}),\\
&(\mathrm{BMKN}) = \frac{1}{2} r^2 (2\omega),\\
&(BMN) = \frac{1}{2}\frac{d}{2} 2r \sin\omega = \frac{1}{2} dr \sin\omega.
\end{aligned}
$$

From the Cosines' Law in ABM, it is:

$$r^2 + d^2 - 2rd\cos\omega = r^2 \Rightarrow \omega = a\cos(\frac{d}{2r}).$$

Combining the above we have

$$(\mathrm{MHNK}) = \pi r^2 \text{ - } 2\mathrm{r}^2 \mathrm{acos}(\frac{\mathrm{d}}{2\mathrm{r}}) + \mathrm{drsin}[\mathrm{acos}(\frac{\mathrm{d}}{2\mathrm{r}})].$$

For $\mathrm{d} = 2\mathrm{r}/3$, the above equation becomes $\mathrm{E2} = (\mathrm{MHNK}) = 1.3066\mathrm{r}^2$, and the fraction $\mathrm{E2/E}$ is

$$E_2 / E = \frac{1.3306 r^2}{\pi r^2} = 0.42.$$

Modeling and Performance Evaluation of SCTP as Transport Protocol for Firewall Control

Sebastian Kiesel and Michael Scharf

Institute of Communication Networks and Computer Engineering,
University of Stuttgart, Germany
{kiesel, scharf}@ikr.uni-stuttgart.de

Abstract. Firewalls are a crucial building block for securing IP networks. The usage of out-of-band-signaling protocols (such as SIP) for VoIP and multimedia applications requires a dynamic control of these firewalls, which can be implemented using the Simple Middlebox Configuration Protocol (SIMCO). In this paper, we study the performance of SCTP and TCP as transport protocols for the transaction-based signaling protocol SIMCO, which requires small end-to-end delays. We present an analytical model in order to quantify the impact of head-of-line blocking in SCTP. Both, the model and measurements reveal that SCTP can significantly reduce the SIMCO response times by leveraging transmission over multiple parallel streams. While a few SCTP streams can almost completely avoid head-of-line blocking, our measurements show that TCP may suffer from rather large end-to-end delays.

1 Introduction

For quite a long time, firewalls have been in use to protect private networks from unwanted access from the Internet. The simplest type are packet filters.

So-called "Next Generation Networks" (NGN) are an emerging technology intended to replace the ISDN based telephone networks by Session Initiation Protocol (SIP) based "Voice over IP" (VoIP) technology in the future. They differ from traditional IP networks by deploying stateful application layer entities such as SIP proxies in the core network. Because of higher security requirements, firewalls are not only used as a customer premises equipment, but also for screening at the interconnection of different operator's networks. Due to the dynamic nature of SIP, these firewalls have to take part in the session signaling. As traffic of many simultaneous calls has to be inspected, performance is an important issue, in particular the call setup delay to which the dynamic configuration of the firewalls is a contributing factor.

The Stream Control Transmission Protocol (SCTP) has been designed as a transport layer protocol especially for signaling applications. Compared to TCP, SCTP adds a multiplexing layer – the so-called streams – on top of its associations. As in-oder delivery of messages is only guaranteed within the same stream, the delaying effect of so-called "head-of-line blocking" can be reduced if the application software can make use of several streams. In this paper, we show how this feature can be used in time-critical firewall control signaling applications.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 451–462, 2006.

While some studies on SCTP performance exist, to the best of our knowledge the end-to-end delay of signaling messages over multiple streams has not been addressed so far. We present an analytical model and measurement results to quantify the impact of head-of-line blocking in SCTP. A related work [1] presents simulation results for the transmission delay of SIP messages transported over UDP, TCP, or SCTP, respectively. Unlike our work, this study only uses one SCTP stream with reliable unordered service and leaves message reordering up to SIP. A recent work [2] analyzes the usage of SCTP for a parallel computing message passing middleware and shows that multiple streams can improve the transmission delays. However, this work is based on measurements only and uses messages that are orders of magnitude larger than typical signaling data.

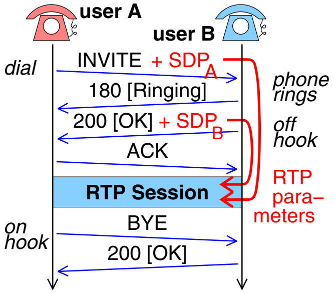
The remainder of the paper is organized as follows. In Section 2, we discuss the interaction of SIP-based VoIP signaling and firewalls. We introduce the IETF MIDCOM architecture and the SIMCO protocol that can be used for firewall control. Section 3 discusses which transport protocols can be used for SIMCO. We give a brief overview of SCTP and present the basic idea how SIMCO can leverage SCTP's multiple streams feature. Section 4 proposes an analytical model of the head-of-line blocking in SCTP for signaling traffic. In Section 5, we present a prototype implementation of "SIMCO over SCTP" and performance measurement results. Finally, Section 6 concludes this paper.

2 Securing IP Telephony Networks by Firewalls

2.1 Firewall Policies and Out-of-Band Signaling

A "firewall" is one or a group of network elements enforcing an access control policy on the traffic at the border between network domains with different security levels and requirements. The access control and forwarding functions can be implemented on different layers of the IP protocol stack, e. g., in the application layer by so-called *proxies*. In contrast, *packet filters* are routers that decide whether to forward a packet by comparing header fields of IP and transport layers with their access control lists (ACL). Often, a policy with respect to application layer services can be implemented by simply comparing the transport layer destination port numbers with a *static* ACL, because of the *well-known port numbers* concept that most "traditional" Internet services follow.

However, some applications such as many VoIP solutions differ by using different protocols for the signaling and the transport of the actual user data (speech). In the considered scenario, the RTP (Realtime Protocol) media stream parameters such as codec and bit rate as well as IP addresses and UDP port numbers are *dynamically* negotiated using SDP (Session Description Protocol) messages embedded in the SIP signaling (Fig. 1). Therefore, a static packet filter configuration would either block all RTP streams or would have to allow all UDP traffic, rendering the firewall's protection almost useless [3]. As signaling messages and media streams may travel on different paths through the network (e. g., to roaming users, see Fig. 2), firewall traversal becomes even more difficult.



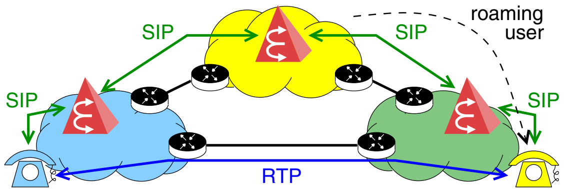


Fig. 1. Negotiation of RTP parameters by means of SIP/SDP

Fig. 2. Signaling messages and media streams may travel on different paths through the network

2.2 Architectures for Firewall Control – IETF MIDCOM

There are several approaches to solve the SIP/RTP firewall traversal problem [3]. A solution is to dynamically add rules to the packet filter by means of a signaling protocol. This can be done in two ways. *Path-coupled* firewall signaling such as the IETF NSIS architecture [RFC 4080] sends messages along the future media path, which request to open so-called "pinholes" in all packet filters on the path.

In contrast, when using *path-decoupled* signaling, SIP messages are sent via SIP entities such as back-to-back user agents (B2BUA, i. e., call-stateful SIP proxies), which control the packet filters on the media path. For the signaling between B2BUA and packet filter, the IETF MIDCOM (MIDdlebox COMmunication) architecture [RFC 3303] can be used. The term "middlebox" refers to the generalized concept of network elements that perform "functions other than the normal, standard functions of an IP router" [RFC 3234], such as packet filters and network address translators (NAT).

Fig. 3 shows one possible MIDCOM application scenario with SIP. The firewall protecting domain "A" consists of the packet filter and the B2BUA. A static rule in the packet filter allows SIP signaling to be sent via the B2BUA. Once the B2BUA has decided to allow the establishment of a specific call, it extracts the RTP parameters from the SIP/SDP messages. It sends "Policy Enable Rule" (PER) requests to the the packet filter in order to open two corresponding "pinholes" (one per direction) for the call duration. The interworking of SIP and

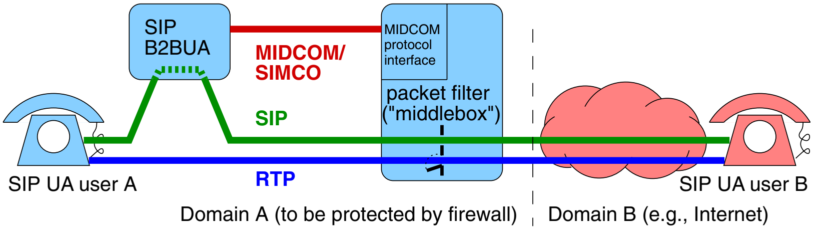

SIP B2BUA
MIDCOM/ SIMCO
MIDCOM protocol interface
packet filter ("middlebox")
SIP
RTP
SIP UA user A
SIP UA user B
Domain A (to be protected by firewall)
Domain B (e.g., Internet)

Fig. 3. The MIDCOM architecture

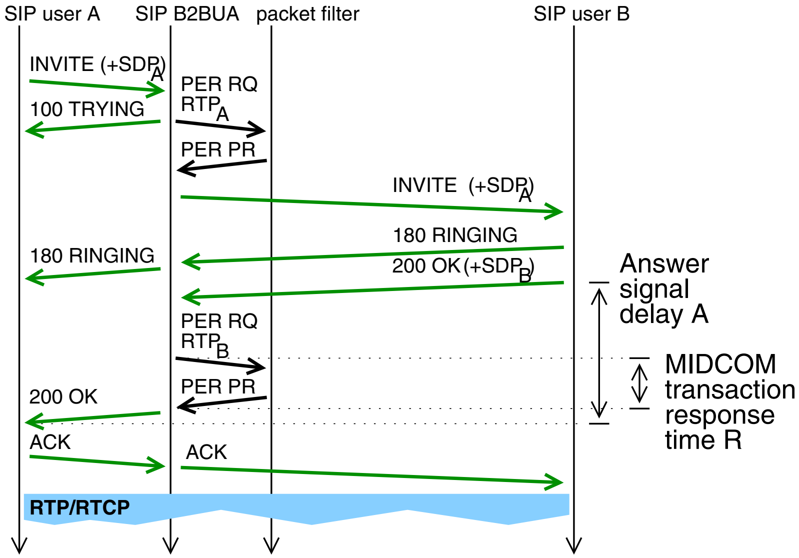


Fig. 4. Interworking of SIP and MIDCOM/SIMCO signaling

MIDCOM signaling is illustrated by Fig. 4. A more detailed description including considerations of the behavior under error conditions can be found in [3].

2.3 SIMCO

MIDCOM is not a specific protocol but a framework architecture, including an abstract protocol semantics [RFC 3989] that can be implemented in several ways, e. g., by means of a suitably crafted SNMPv3 MIB. An alternative approach is the SIMCO (SImple Middlebox COnfiguration) protocol [4], a transaction based protocol using simple binary TLV message encoding. A transaction consists of a request from the SIMCO agent (e. g., embedded in the SIP B2BUA) and a positive or negative reply from the middlebox. SIMCO transactions are used to create, modify or delete so-called “policy rules” at the middlebox, which are the generalized concept of pinholes in packet filters, address bindings in NATs, etc. They are described by various address parameters. Policy rules are soft states, i. e., they are associated with a lifetime attribute and will be removed automatically from the middlebox if the lifetime is not refreshed in time.

All SIMCO messages belonging to one transaction are identified by means of a transaction identifier (TID), which is uniquely assigned by the SIMCO agent. When a SIMCO agent asks the middlebox to establish a new policy rule, e. g., by means of a PER (Policy Enable Rule) request, the middlebox creates the rule and assigns a unique policy rule identifier (PID) to it. The PID is returned to the agent, e. g., in the PER positive reply. The agent uses the PID in later transactions, such as PLC (Policy Lifetime Change) for refreshing the softstate or deleting a rule (new lifetime = 0), to refer to this specific policy rule (Fig. 5).

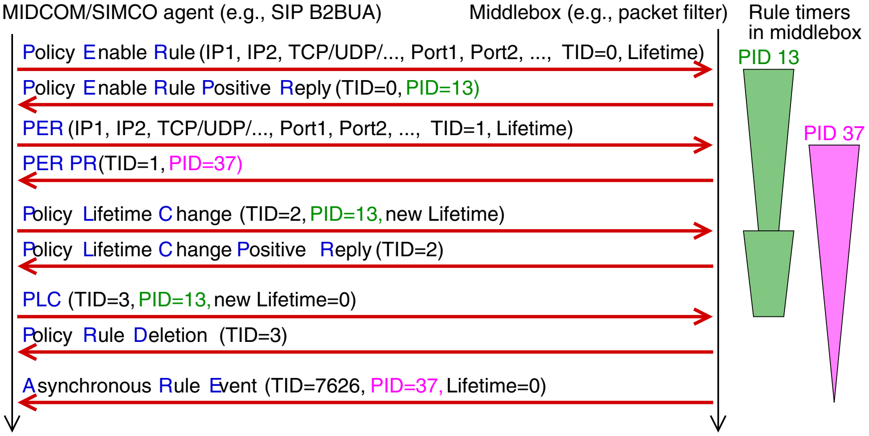

MIDCOM/SIMCO agent (e.g., SIP B2BUA)
Middlebox (e.g., packet filter)
Rule timers in middlebox
Policy Enable Rule (IP1, IP2, TCP/UDP/..., Port1, Port2, ..., TID=0, Lifetime)
Policy Enable Rule Positive Reply (TID=0, PID=13)
PER (IP1, IP2, TCP/UDP/..., Port1, Port2, ..., TID=1, Lifetime)
PER PR(TID=1, PID=37)
Policy Lifetime Change (TID=2, PID=13, new Lifetime)
Policy Lifetime Change Positive Reply (TID=2)
PLC (TID=3, PID=13, new Lifetime=0)
Policy Rule Deletion (TID=3)
Asynchronous Rule Event (TID=7626, PID=37, Lifetime=0)
PID 13
PID 37

Fig. 5. SIMCO transactions create, modify and delete policy rules

In addition to the transactions explained above, SIMCO specifies transactions for the management of a SIMCO association and asynchronous notifications to be sent from the middlebox to the agent, e. g., if a policy rule is deleted because of expired lifetime. The SIMCO specification [4] assumes that all transactions between an agent and a middlebox are transported in one SIMCO association over a single persistent TCP connection. Both are established in advance in order to avoid transaction delays caused by the TCP and SIMCO handshake.

As shown in Fig. 4, the response time R of the second PER transaction contributes to the answer signal delay A, which is inconvenient for the users and should therefore be minimized [5]. R consists of the local processing time in the middlebox plus the message transmission delay. In the remainder of the paper, we will focus on the latter effect and investigate in detail how to minimize the transmission delay by using SCTP as transport protocol for SIMCO.

3 Transport Protocols for Firewall Control

Traditionally, there have been two transport layer protocols in the Internet protocol suite. The Transmission Control Protocol (TCP) provides connection oriented, reliable transmission. It is the default transport protocol for SIMCO. The User Datagram Protocol (UDP) offers connectionless, unreliable transport and is therefore unsuitable for SIMCO. The Stream Control Transmission Protocol (SCTP) has been developed as a third transport layer protocol for IP, especially for signaling applications. It will be introduced briefly in the next section before its applicability for SIMCO will be investigated.

3.1 Stream Control Transmission Protocol (SCTP)

SCTP [RFC 2960] has originally been designed as a part of the SIGTRAN architecture [RFC 2719] for the transport of SS7 [6] PSTN/ISDN telephony signaling

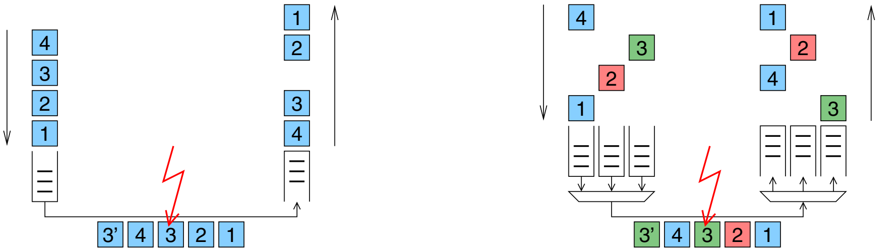


Fig. 6. Illustration of head-of-line blocking: one TCP connection (left) vs. one SCTP association with 3 streams (right)

over IP. While this rather special purpose is achieved by adaptation layers (e. g., M3UA [RFC 3332]) on top of it, SCTP itself has been designed as a generic transport protocol for IP networks, optimized for signaling applications. It has mechanisms for deployment in environments with high reliability and security requirements, such as "multihoming", i. e., support for having endpoints with several physical network interfaces as well as mechanisms to protect from denial-of-service and blind spoofing attacks [7].

With respect to user data transmission, SCTP provides a reliable datagram service. Similar to UDP, SCTP preserves the boundaries of upper layer protocol (ULP) messages. Therefore and different to TCP, no byte counters or frame delimiters are needed in the ULP. Unlike UDP, SCTP detects packet loss, duplicate packets or bit errors and retransmits or discards the respective packets. SCTP also uses flow control and congestion control algorithms similar to those of TCP.

SCTP allows to split one association (SCTP term for connection) into up to 65536 logical subchannels per direction, so-called streams. Each user message is transmitted in one of these streams. SCTP ensures in-order delivery within the same stream. If one message is lost or corrupted in the network and has to be retransmitted, only the corresponding stream is subject to head-of-line blocking whereas messages of other streams can still be delivered. This is illustrated in Fig. 6: Using SCTP, message #4 may be delivered to the ULP before message #3 has been retransmitted, as it is in another stream. Message ordering can even be disabled completely using the "unordered" flag.

Using one association split up into several streams – instead of using multiple associations bearing only one stream each – improves the efficiency of the TCP-like fast retransmit algorithm as it is applied on the aggregate message flow.

3.2 SIMCO over SCTP

When designing "SIMCO over SCTP", a very important problem is how to leverage SCTP's multiple streams feature, in order to reduce the impact of head-of-line blocking. This can be further divided into two questions: First, how many streams to use for good performance results while not wasting resources. This will be investigated in detail in the later sections of this paper.

The other problem is how to distribute SIMCO messages evenly over several streams while retaining causality for SIMCO. It is important that the transport layer protocol preserves the order of transactions that refer to the same policy rule. For example, it would be undesirable if a SIMCO agent requested a policy rule and immediately afterwards canceled it, but the delete message was delivered to the middlebox before the enable message. However, there is no requirement that prohibits reordering of SIMCO messages that refer to different policy rules.

Our basic idea is therefore to have several bidirectional stream pairs within the SCTP association. Two requirements shall be fulfilled: (1) All SIMCO messages that belong to one transaction shall be sent over the same pair. (2) All transactions that refer to the same policy rule shall be sent over the same pair.

As transactions always consist of a request sent by the agent and a reply sent by the middlebox, requirement (1) can be implemented in a stateless fashion: The middlebox sends replies on the same stream number as the corresponding request was received on. Requirement (2) can be fulfilled in the following way: Initially, the agent may use any strategy (e. g. round robin) for choosing the stream number on which a message requesting a new policy rule is to be sent. As described in Section 2.3, the middlebox assigns a unique PID to the new policy rule and returns it to the agent. The mapping from PID to stream number will be stored by the agent and used for sending all subsequent transactions that modify or delete this policy rule. The detailed specification of our approach, including special cases, can be found in [8].

4 A Model for Head-of-Line Blocking in SCTP

In the following, we model the SIMCO response time when SCTP is used as transport protocol. We assume a scenario where the packet filter is located between two large domains, i. e., the busy hour call arrival rate is rather large.

4.1 Workload Model

As shown in Fig. 4, two pinholes are required to establish a call. The number of SIMCO transactions required to open, maintain, and close a pinhole depends on the call duration. According to Fig. 5, the pinhole is opened by a PER request, and a PLC with a lifetime extension of 0 is sent when the call is terminated. Due to additional PLCs during the call, the total number of SIMCO transactions per pinhole is $n(T) = 2 + \lfloor \frac{T}{L} \rfloor$, where T is the call duration and L the lifetime extension period. The overall rate of SIMCO transactions depends on the call duration distribution $f(T)$ and the rate of pinhole opening requests λ:

$$\lambda_{\mathrm{SIMCO}} = \lambda \cdot \int_0^\infty n(T) \cdot f(T) \,\mathrm{d}T \,. \tag{1}$$

If we assume that the call duration is exponentially distributed with mean h and PDF $f(T) = \frac{1}{h} \exp(-\frac{T}{h})$, the mean inter-arrival time (IAT) d of SIMCO messages in one direction can be approximated as $d = \frac{1}{\lambda_{\mathrm{SIMCO}}} \approx \frac{1}{\lambda} \left(\frac{3}{2} + \frac{h}{L}\right)^{-1}$.

4.2 Resequencing Delay over Multiple SCTP Streams

In this section, we model the effect of head-of-line blocking when several SCTP streams are used in parallel for data transmission, i. e., the SIMCO traffic is equally distributed over $N \geq 1$ streams. We assume that the path between the two endpoints has a constant unidirectional delay of Δ and thus a minimum round-trip time $RTT = 2\,\Delta$. The path is supposed to suffer from symmetric random packet losses with loss probability p, which may be caused for instance by congestion or transmission errors. Of course, for a well-dimensioned signaling network p is likely to be small. Still, it is important to quantify the performance impact of lossy links in order to derive system dimensioning guidelines.

Due to the packet loss in both directions, an acknowledgement for a DATA chunk arrives at the sender with probability $p_S = (1-p)^2$. An endpoint can detect packet loss if transmission sequence numbers (TSNs) are missing in the selective acknowledgements (SACKs). A SACK, which is sent upon the reception of a DATA chunk on one stream, contains missing TSN reports for all streams. Similar to the "fast retransmit" mechanism in TCP, an SCTP endpoint retransmits data when three subsequent SACKs include a missing report [9]. The reliable data delivery is also ensured by a timeout mechanism. However, this mechanism is usually only required if multiple packets get lost in sequence.

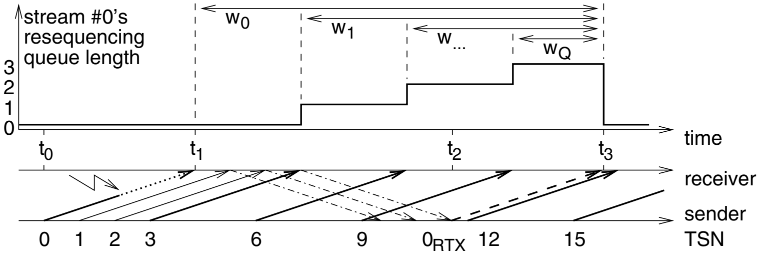


Fig. 7. Illustration of resequencing delays for 3 SCTP streams

The SCTP error recovery by a fast retransmit is illustrated in Fig. 7. For this figure, we assume that the SCTP association has $N = 3$ streams and a round robin scheduling strategy is applied, i. e., the DATA chunks with transmission sequence numbers 0, 3, 6, ... are sent via stream #0, while DATA chunks with TSNs 1, 4, 7, ... and 2, 5, 8, ... are sent via streams #1 and #2, respectively. For simplicity, DATA chunks are supposed to be sent with constant IAT d. Furthermore, we assume that the sending window does not restrict the amount of DATA chunks sent, which is reasonable if the packet loss probability is small.

In this example, the DATA chunk with TSN 0 is lost. t_0 denotes the point in time when this packet is sent, t_1 is when it should arrive at the receiver. At $t_2 = t_0 + RTT + 3\,d$ the sender has received 3 SACK chunks with missing reports and performs the retransmission. Note that SCTP's SACK messages contain information about missing DATA chunks for all streams. When the retransmitted packet arrives at the receiver at $t_3 = t_2 + RTT/2$, all DATA chunks in stream #0's resequencing queue can be delivered to the upper layer protocol entity.

The waiting times w_n of DATA chunks in the resequencing queue depend on the time $D = w_0$ to detect and recover from the packet loss. As shown by Fig. 7, the minimum value for D is $RTT + 3\,d$. However, D may be larger if SACKs get lost, too. Each DATA chunk triggers a SACK chunk, but both may get lost. The probability that three SACKs arrive at the sender, after i DATA chunks have been sent, is $P(i) = p_{\mathrm{S}}^3\,(1-p_{\mathrm{S}})^{i-3}\binom{i-1}{i-3}$. From this follows

$$D \approx RTT + d\sum_{i=3}^{\infty} P(i)\,i = RTT + \frac{3\,d}{(1-p)^2}\ . \tag{2}$$

This expression is an approximation only since the retransmission may get lost, too. In this case, a retransmission timeout is required which may further enlarge the recovery period. Several subsequent lost DATA chunks may also trigger overlapping fast recovery periods, which are difficult to describe by a simple model. We neglect both effects in this model since they hardly occur if the packet loss probability p is small.

The number of DATA chunks that have to be queued until the retransmission arrives is $Q = \lfloor \frac{D}{d\,N} \rfloor$. The resequencing delay of the first DATA chunk after the lost one is given by $w_1 = D - N\,d$. The subsequent waiting times are $w_2 = D - 2\,N\,d$, $\ldots$, $w_Q = D - Q\,N\,d$. The mean waiting time is the sum of all w_i divided by the mean number of DATA chunks between two losses, which is $1/p$. The mean increase of the unidirectional end-to-end delivery delay is thus

$$W = p\sum_{i=0}^{Q} w_i = p\left((Q+1)\cdot D - \tfrac{Q\,(Q+1)}{2}\,N\,d\right)\ . \tag{3}$$

For bidirectional transactions as in the case of SIMCO, head-of-line blocking may occur in both directions and the mean response time thus follows as

$$R = RTT + 2\,W + \delta\ , \tag{4}$$

where δ represents the processing time in the end systems. As already mentioned, R must be small to minimize the answer signal delay perceived by users.

The remaining question is the optimal number of SCTP streams. Using a large number of streams may not be efficient since this may waste resources (memory) in the endpoints. Under the assumption that for small values of p at most one stream is blocked, head-of-line blocking can be avoided completely if no DATA chunks get queued before the retransmission is triggered, i. e., $Q = 0$. This is fullfilled for $N \geq M$ with $M = \lceil \frac{D}{d} \rceil$. According to (2), M is quite insensitive to p. From this follows the optimal number of streams as

$$M \approx \left\lceil RTT \cdot \lambda \cdot \left(\tfrac{3}{2} + \tfrac{h}{L}\right) + 3 \right\rceil\ . \tag{5}$$

5 Performance Evaluation

5.1 Measurement Setup

In order to evaluate our "SIMCO over SCTP" specification [8], a prototype compliant to [4, 8] has been implemented [10]. As shown in Fig. 8, the middlebox software can control a packet filter (Linux Netfilter). For functional tests, the SIMCO agent has been integrated into the VOVIDA SIP back-to-back user agent (B2BUA) [11, 3]. A load generator emulating user behavior has been implemented for measuring the SIMCO transaction response time (see Fig. 9).

The SIMCO software was implemented in C++ for Linux (kernel 2.6.11) and Solaris 10. The Linux version can use either the "lksctp"-kernel module or standard Linux TCP (using SACKs). For both protocols the "nodelay" socket options have been enabled. For Solaris, so far only TCP performance has been investigated. Measurements were made using two 2.4 GHz Pentium 4 or 500 MHz UltraSPARC IIe computers connected by 100 Mbps Ethernet to a network emulator, which adds a delay of $\Delta = 10\,\text{ms}$ in each direction and randomly drops IP packets with given probability p. The interaction of the middlebox software with the packet filter was disabled to isolate the measured delay from this overhead.

In the following we present the measurement results for one typical scenario with a large-scale softswitch. The load generator requests pinhole openings with exponential IAT $\frac{1}{\lambda} = 30\,\text{ms}$ and exponential lifetime $h = 180\,\text{s}$. With two pinholes per call and a typical busy hour load of $\rho = 0.05$ Erlang, this corresponds to a mean number of $m = \frac{1}{2}\,h\,\lambda = 3,000$ simultaneous calls and $S = \frac{h\,\lambda}{2\,\rho} = 60,000$ subscribers. The rule softstates are refreshed every $L = 120\,\text{s}$. From this follows $d \approx 10\,\text{ms}$ as mean IAT of SIMCO messages. Measurements with other parametrizations revealed similar results, which are documented in [12].

5.2 Measurement Results

Fig. 10 shows the mean SIMCO response time as a function of the packet loss probability p. All values have been obtained by averaging over the response time of PER requests during a measurement period of 1000 s after the load generator has reached the steady state. Fig. 10 reveals that using more than one SCTP

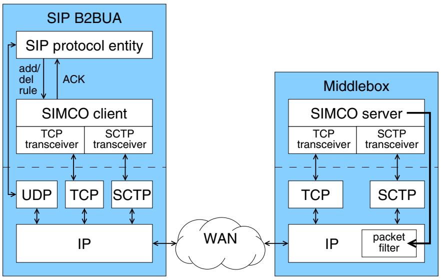


Fig. 8. SIMCO/SCTP prototype with B2BUA for proof of concept in SIP testbed

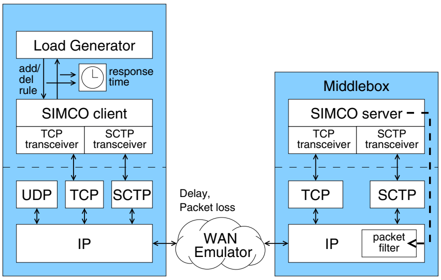


Fig. 9. SIMCO/SCTP testbed with load generator for performance measurements

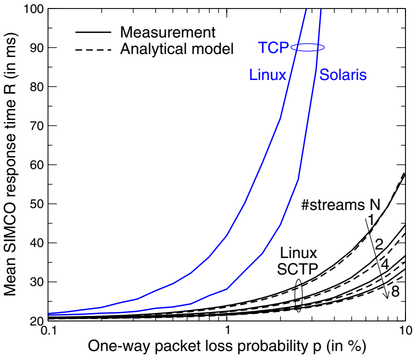


Fig. 10. Comparison of SCTP/TCP

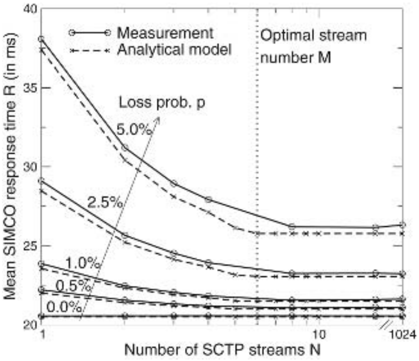


Fig. 11. Impact of SCTP streams no

stream can significantly improve the SIMCO response time R even for moderate loss probabilities such as $p = 2\,\%$. The difference gets larger for higher p, but such situations will hardly occur in well-dimensioned signaling networks.

Fig. 11 presents the SCTP measurement results as a function of the number of streams N. They match very well the response time predicted by the analytical model in eq. (4), with a processing delay assumed to be $\delta = 0.5\,\text{ms}$. The model slightly underestimates the response time for $p > 1\,\%$. This is probably due to the impact of multiple fast retransmits and timeouts that cannot be neglected for high loss probabilities. Fig. 11 also confirms that using a value N larger than the optimum value (here: $M = 6$) does not significantly improve performance.

Futhermore, Fig. 10 presents measurement results for TCP, both for Linux and Solaris operating systems. In theory, one would assume that TCP has a similar performance like SCTP with one stream. However, according to our measurements the mean SIMCO response time is significantly larger. Even worse, TCP is not able to transport the offered load of about 100 transactions/s for packet loss probabilities larger than $7\,\%$, which is manifested by socket buffer overflows. In particular the Linux TCP implementation, which is known to be highly optimized, performes quite bad even for packet loss probabilities much smaller than $1\,\%$. Furthermore, there is a non-negligible probability for high call-setup delays. For example, the $99\,\%$ quantile of the SIMCO response time is ca. 100 ms for $p = 1\,\%$ if Linux TCP is used. Our analytical model for the head-of-line blocking does not explain this TCP-specific effect.

We have verified the measurements with a pair of simple test programs that use straightforward socket calls. These tests confirm the difference between TCP and SCTP. To the best of our knowledge, this effect has not been reported so far. An analysis of TCP traces shows that sometimes data arriving from the application layer is not immediately sent to the network. These additional delays could be caused by the TCP congestion control that reduces the sending window when facing packet loss. A more detailed insight into this effect can probably be obtained by means of simulation, but this is left for further study.

6 Conclusions

In this paper, we study the performance of the SIMCO protocol using SCTP and TCP as transport protocols. Being a typical signaling protocol, SIMCO can benefit from protocol mechanisms for high-reliability environments, which SCTP provides. Compared to TCP, SCTP can also significantly reduce the SIMCO response time by leveraging transmission over multiple streams, which reduces head-of-line blocking. We propose an analytical model to quantify this effect, and verify it with measurements. We show that a small number of SCTP streams is sufficient to almost completely avoid head-of-line blocking. Furthermore, our measurements, both for Linux and Solaris, reveal that using TCP for transaction-based signaling causes significant delays even for small packet loss probabilities.

Acknowledgements

The authors would like to thank Christian Blankenhorn, Sebastian Beutel, and Thomas Ruschival for their help with the testbed and the measurements, as well as Martin Stiemerling and Michael Tüxen for valuable discussions and comments, especially on the IETF documents. Michael Scharf is funded by the German Research Foundation (DFG) through the Center of Excellence (SFB) 627.

References

1. G. Camarillo, R. Kantola, and H. Schulzrinne, "Evaluation of Transport Protocols for the Session Initiation Protocol," *IEEE Network*, vol. 17, no. 5, 2003.
2. H. Kamal, B. Penoff, and A. Wagner, "SCTP versus TCP for MPI," in *Proc. Supercomputing 2005*, Seattle, USA, Nov. 2005.
3. A. Müller and S. Kiesel, "Issues with the Interworking of Application Layer Protocols and the MIDCOM Architecture," in *Proc. Eunice Summer School*, 2004.
4. M. Stiemerling, J. Quittek, and C. Cadar, "Simple Middlebox Configuration (SIMCO) Protocol Version 3.0," IETF draft - work in progress, May 2005.
5. ITU-T Study Group 2, "Network grade of service parameters and target values for circuit-switched services in the evolving ISDN," ITU-T, Rec. E.721, May 1999.
6. ITU-T Study Group XI, "INTRODUCTION TO CCITT SIGNALLING SYSTEM No. 7," ITU-T, Recommendation Q.700, Mar. 1993.
7. S. Kiesel, "On the Use of Cryptographic Cookies for Transport Layer Connection Establishment," in *Proc. EUNICE Summer School*, 2002.
8. S. Kiesel, "SIMCO over SCTP," IETF draft - work in progress, Oct. 2005.
9. R. Stewart, "Stream Control Transmission Protocol (SCTP) Specification Errata and Issues," IETF draft - work in progress, Oct. 2005.
10. C. Blankenhorn, "Evaluation of SCTP as Transport Protocol for Transaction-based Applications at the Example of a Protocol for Firewall Control," Student project (in German), University of Stuttgart, IKR, 2005.
11. A. Müller, "Extension of a SIP proxy by security functions," Student project (in German), University of Stuttgart, IKR, 2004.
12. S. Kiesel, M. Scharf, S. Beutel, and T. Ruschival, "Performance Measurement Results of SIMCO over TCP and SCTP," University of Stuttgart, IKR, Internal Report 53, 2006.

Transport Layer Issues in Delay Tolerant Mobile Networks

Khaled A. Harras and Kevin C. Almeroth

Department of Computer Science,
University of California, Santa Barbara,
Santa Barbara, CA 93106-5110
{kharras, almeroth}@cs.ucsb.edu

Abstract. The tremendous increase in wireless devices and user mobility have ultimately resulted in a new set of networking challenges that previously did not exist. Some of these challenges include large delays, intermittent connectivity and most importantly, the absence of an end-to-end path from sources to destinations. Networks characterized by one or more of these challenges are called *Delay Tolerant Networks (DTNs)*. Researchers have studied DTNs with a major focus on routing issues in such extreme environments. As a result, in this paper, we shift this focus towards addressing and studying transport layer issues in extreme networking environments. We particularly concentrate on investigating and comparing several reliability approaches in a specific category of DTNs known as Delay Tolerant Mobile Networks (DTMNs). We present four different reliability approaches in DTMNs. We also evaluate these approaches under various network conditions via simulation. Our goals from this study are to examine the impact of these reliability approaches, understand the tradeoffs between them, and open the way for further work in transport layer issues in delay tolerant networks.

Keywords: Delay Tolerant Networks, Mobile Networks, Reliability.

1 Introduction

With the explosive evolution in wireless devices, many new network environments have emerged. Some of these environments include, satellite and interplanetary [7], military/tactical [11], disconnected remote village [13], and disaster rescue [1] networks. These new environments have become more prominent with recent natural disasters. The need to establish communication to serve applications that run in such extreme environments has never been more evident.

The emergence of these new environments has lead to a new set of networking challenges. Some of these challenges include network partitioning, large delays, intermittent connectivity, high link error rates, and heterogeneous underlying networks and protocols. As a result, a new set of assumptions needs to be considered, such as large delays, intermittent connectivity, and most importantly, the absence of an end-to-end path from a source to a destination.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 463–475, 2006.

These new challenges and assumptions have spurred much research in such extreme and mobile environments. Researchers in Mobile Ad Hoc NETworks (MANETs) have tackled mobility problems with a major focus on routing [9], [14], [15], [16]. MANETs, however, fail to address all of the emerging challenges listed above, since they only consider scenarios where an end-to-end path exists from a source to a destination. Other research has started to address the challenge of communicating even though such a path does not exist. This research includes disconnected mobile networks [12], [18], sparse sensor networks [10], [17], and different forms of Delay Tolerant Networks (DTNs) [5], [4], [8], [19], [20], [6]. These areas have introduced different DTN architectures and solutions with a focus on solving *routing* and *message delivery* problems in such extreme environments.

With the work in DTNs mainly focused on routing, we shift our focus towards studying transport layer issues. Most of the services offered by existing transport layer protocols, such as TCP, have been overlooked. In general, the most important services offered by TCP are ports, connections, sequencing, congestion control, and reliability. Some of these services are easy to deploy in DTNs, while others require further research. We briefly look at each of the TCP-style transport functions in DTN environments.

Of the TCP services previously mentioned, ports are still provided and used by overlay protocols for communication in DTN environments. Next, sequencing is done the same way as in TCP, with the exception that sequence numbers are assigned to *message bundles* rather than to individual packets. Connection establishment, on the other hand, is impossible in such environments due to the primary assumption of the absence of an end-to-end connection. The only remaining services, therefore, are congestion control and reliability. Congestion control is a more challenging function to deploy because propagating live congestion-related information across DTN environments is hard. This difficulty is due to the unstable nature of DTN environments. Addressing congestion control is left to future work. We are now left to focus on reliability, a service critical to many of the applications that run in DTN environments.

In this paper, we introduce four different end-to-end reliability approaches for a specific DTN architecture, known as Delay Tolerant Mobile Networks (DTMNs), which are large-scale disconnected mobile networks [6]. First, *hop-by-hop* reliability depends only on sending acknowledgments along every hop in the path. Second, *active receipt* achieves reliability by delivering an *active* end-to-end acknowledgment over the DTMN. Third, *passive receipt* reliability implicitly sends an end-to-end acknowledgment through the network. Fourth, *network-bridged receipt* sends an acknowledgment over another network that exists in parallel to the DTMN. With the multiple devices people currently carry, we can use other parallel networks, such as cell networks, as network bridges to transmit acknowledgements or other control-related information. We evaluate these reliability approaches in DTMNs under various network conditions via simulations. Our goals in this study are to examine the impact of these reliability approaches, understand the tradeoffs between them, and open the way for further work in transport layer issues in delay tolerant networks.

The remainder of this paper is organized as follows. Section 2 first introduces related work. Section 3 then gives an overview of DTMNs. We discuss the different reliability approaches in Section 4. The simulation environment and results are described in Section 5. Finally, we conclude in Section 6.

2 Related Work

Research in the areas of MANETs [9], [14], [15], [16], disconnected mobile networks [12], [20], [19], sparse sensor networks [17], and delay tolerant networks (DTNs) [4], [8], [6], have addressed issues related to the challenges stated in Section 1. We briefly present some of the solutions in these areas.

Work in MANETs has mainly focused on routing, introducing various protocols that find end-to-end paths between nodes [9], [14], [15], [16]. Since such paths mostly do not exist in the applications with which we are concerned, MANETs, therefore, fail to address the transport challenges we address in this paper.

Most of the solutions presented by disconnected mobile and sparse sensor networks rely on some form of store-and-forward relaying of messages. This relaying includes different message delivery techniques; the differences are in the underlying assumptions over which they operate. For example, some solutions assume full control over node movement [12]. Others, such as message ferrying, assume knowing the path that some nodes will take and the time at which these node will take that path [20]. Some consider using *data mules* to gather data from static sensors [17], while others find optimal paths for *ferries* to deliver messages between sparse static nodes [19]. Epidemic Routing, on the other hand, simply floods the network to ensure message delivery [18]. Our previous work provides different approaches to control these floods [6]. All of these solutions fundamentally focus on message delivery and routing techniques in challenged extreme environments. To the best of our knowledge, no existing work thoroughly studies transport layer issues in such environments.

With respect to DTNs, members of the Delay Tolerant Networking Research Group (DTNRG) [3] introduce an architecture that helps achieve connectivity among heterogeneous networks in extreme environments [2], [4]. A *bundle layer protocol* is introduced to handle many of the challenges previously discussed using a store-and-forward approach. They also propose the idea of *custody transfer*, where a *custodian* assumes the responsibility of reliably delivering a bundle to the next custodian on the path to the destination [5]. Jain et al. expand on the DTN work by studying routing issues in such extreme environments. Again, the focus in the DTN work is almost exclusively on routing [8]. The DTN community has briefly addressed reliability through custody transfer [5] in the bundle layer protocol [4]. However, there has been no in-depth study or evaluation of its performance, especially when compared to other approaches. In this paper, we examine this approach, along with others that we propose, particularly over delay tolerant mobile networks (DTMNs).

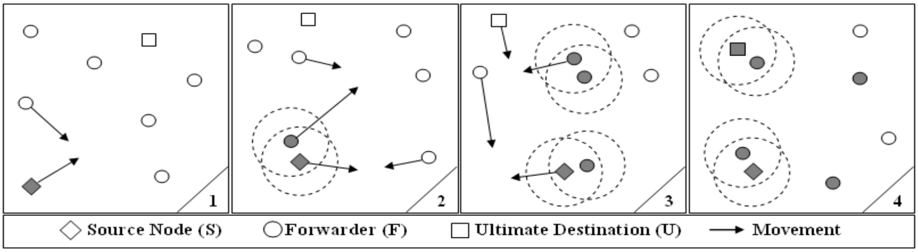


Fig. 1. An example of message delivery in DTMNs. *Infected* nodes are shaded.

3 An Overview on DTMNs

The work presented in this paper uses DTMNs [6] as the underlying network environment. Since this is the environment we use to study our reliability approaches, we give a brief overview of DTMNs' basic architecture and terminology.

DTMNs are a special kind of DTNs with the assumption that all nodes in the network are mobile, and that end-to-end paths may not exist between any two nodes in the network. In this environment, due to the sparseness and mobility of nodes, each node is viewed as a "region" with respect to the classical DTN architecture [4]. Similarly, each node acts as a DTN gateway to perform overlay bundle relaying of messages. There are two key assumptions in DTMNs with respect to network nodes. First, nodes are *blind*. They do not know any information regarding the state, location, or mobility patterns of other nodes. Second, nodes are *autonomous*. Each node has independent control over itself and its movement.

We now show the operation of DTMNs, an example of which is illustrated in Figure 1. The number in the bottom right corner of each sub-figure represents the sequence of snapshots taken for a DTMN. The figure shows the basic method for propagating messages through the network from the source node, *S*, to the ultimate destination, *U*, with the aid of other forwarder nodes, *F*. Shaded nodes are what we refer to as *infected nodes*, nodes which have received a copy of the message. All infected nodes, including the source, try to infect other nodes at varying degrees of *willingness*. This willingness is generally an indication of how hard a given node tries to forward to, or infect, other nodes.

We note that a DTMN could be viewed both as a full DTN in itself, where each node is both a region and a DTN gateway, or as a single region within the classical DTN architecture [4]. Due to this vagueness, we study the reliability approaches only over DTMN environments in order to focus on the performance and tradeoffs between these approaches. We believe, however, that the results of our work will help us better understand reliability challenges in DTNs in general.

4 Reliability Approaches

We present in this section the four reliability approaches that we study in this paper. First, we discuss the most basic reliability approach for DTMNs, which is *hop-by-hop*. Afterwards, we talk about two different approaches for delivering an end-to-end acknowledgement over a DTMN. These approaches are *active receipt* and *passive receipt*. Finally, we propose a novel modification to the typical DTMN architecture by introducing the idea of *network-bridged receipt*.

4.1 Hop-by-Hop

Hop-by-hop reliability was first introduced in classical DTNs [4]. The idea there, however, was to deliver a message across a given region on the path to the destination, where each region represents a hop. Gateways at the edges of these regions act as custodians and take the responsibility of reliably delivering message bundles across the region [5]. Therefore, there is no end-to-end acknowledgment in these cases; the source only knows whether the next gateway received the message or not, and assumes the gateway will take care of the rest. We build on this idea, and use it as the base reliability approach for DTMNs.

We apply hop-by-hop reliability, however, differently in DTMNs. With the extreme hostility and mobility assumed in DTMN applications, each node in the network acts as a region *and* a gateway with respect to the DTN architecture. Therefore, any exchange of messages between nodes is acknowledged, and all nodes are assumed to reliably forward the message.

The operation of hop-by-hop reliability in DTMNs is illustrated in Figure 2. The source, S, sends a message, M, to the ultimate destination, U, with the aid of forwarder nodes, F. Each time M is *successfully* delivered to any node, an acknowledgment, A, is then sent back to acknowledge the receipt of M. The forwarder nodes along with the source node try to infect as many nodes as possible according to their willingness level. Given enough time and mobility, S assumes that M will eventually reach U. Even though hop-by-hop does not ensure end-to-end reliability, it has the advantage of minimizing the amount of time M remains in S's buffer. This is because S does not need to wait for any end-to-end acknowledgment. We use hop-by-hop as the base approach over which we build the other end-to-end reliability approaches.

4.2 Active Receipt

While the hop-by-hop approach ensures some level of reliability, it does not ensure end-to-end reliability. This limitation could be a problem in cases where

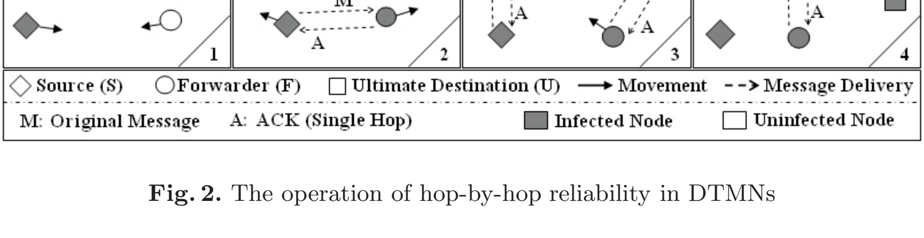


Fig. 2. The operation of hop-by-hop reliability in DTMNs

failures, such as the destruction of a node in a battlefield, or the breakdown of a node in a disaster rescue operation, are likely to occur. In such cases, some form of added end-to-end reliability is required. We overcome this drawback of the hop-by-hop approach by introducing the *active receipt*.

Active receipt is basically an end-to-end acknowledgment, which we call a *receipt*, created by U after it receives M from S. This receipt is *actively* sent back to S. By "actively", we mean that nodes treat this receipt as a new message that needs to be forwarded.

We demonstrate the operation of active receipt in Figure 3(a). The first snapshot starts at the time when U has just received M, with most of the F nodes already infected with M. U then creates the active receipt, R, which is forwarded through the F nodes until it reaches S, shown in the third snapshot of Figure 3(a). Throughout this process, we observe how R *cures* the infected nodes in the network by stopping their transmission of M. R is also cached according to the nodes' willingness levels to prevent re-infection of M. Even though this cure eventually stops the epidemic spread of M through the network, R itself starts to spread epidemically until some timeout or TTL value. The cost of carrying and transmitting R, however, is less than M due to the small size of R.

4.3 Passive Receipt

While active receipt offers end-to-end reliability, its cost in many situations is high. This high cost is because active receipt reaches a point where two messages, rather than one, are infecting nodes in the network. Therefore, we introduce *passive receipt*, which ensures end-to-end reliability, without the incurred cost of active receipt. The idea is to have an implicit/passive receipt, instead of an active one, traverse the network back to S.

We use Figure 3(b) to help clarify the operation of passive receipt. The first snapshot, similar to Figure 3(a), starts at the time when U just received M.

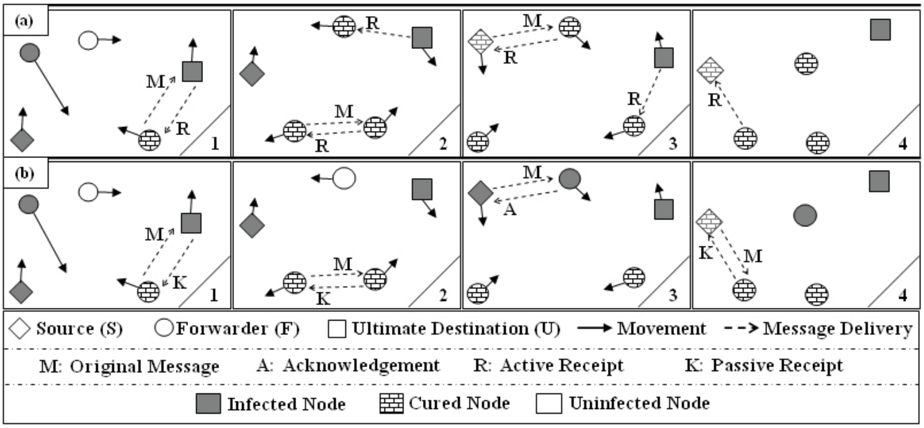


Fig. 3. Demonstrating and comparing (a) active receipt and (b) passive receipt reliability approaches in DTMNs

However, instead of generating a new active receipt, R, an implicit kill message, K, is sent to the infected node to stop it from sending M. The idea is that K is sent by the cured nodes (or U) *only* when they are encountered by one of the infected nodes trying to pass M on to them. In other words, cured nodes do not actively send K messages, they simply wait for active infected nodes to come in their way and stop them from sending M.

The operation of the passive receipt is better understood when compared to active receipt, as illustrated in Figure 3. The first difference is shown in both second snapshots, where in Figure 3(a), R is actively sent to an infected as well as an uninfected node. In the case of passive receipt shown in second snapshot of Figure 3(b), however, K is only sent to the infected node *after* this infected node had tried to pass M to a cured node.

This reduction in cost introduced by the passive receipt approach is not free when compared to active receipt. Even though an end-to-end receipt is received by S in both cases, S receives the end-to-end receipt more rapidly in the case of active receipt. When using the passive receipt, K is received by S at the fourth snapshot, as opposed to receiving R at the third snapshot using active receipt. The reason for this difference in receipt arrival time is that with the active approach, R spreads rapidly in the network, which helps it reach S more quickly than the passively spreading K. This passiveness also results in having infected nodes in the network take a longer time to be cured, as shown in the fourth snapshot in Figure 3(b). This means that the chances of having some infected nodes still trying to send M after S received a receipt, is higher in the passive receipt approach than the active receipt.

4.4 Network-Bridged Receipt

We now introduce a new assumption to the DTMN architecture that enables us to create another reliability approach. This assumption is based on the widespread use of cell phones. We propose exploiting the availability of the cell network by using it as an alternative path for our communication protocol. While such a network does not have the required bandwidth for delivering large amounts of data, it *could be* used for transmitting lightweight control information. Therefore, we use this cell network only for transmitting an end-to-end receipt from the destination back to the source.

This idea is illustrated in Figure 4. We note that all nodes in the network are capable of mobility, however, for clarity, we do not include mobility in the figure. The cell network acts as a bridge between nodes in the DTMN. The cell network is characterized by its continuous end-to-end, low bandwidth connections. The DTMN network, on the other hand, is characterized by its discontinuous non-end-to-end, high bandwidth. In such a setup, large messages, M, are typically transmitted from S over the DTMN using the base hop-by-hop reliability approach until it reaches U. The end-to-end network-bridged receipt, R, would then be transmitted over the cell network instead of the DTMN. If we assume that other nodes in the network also have access to the cell network, R could

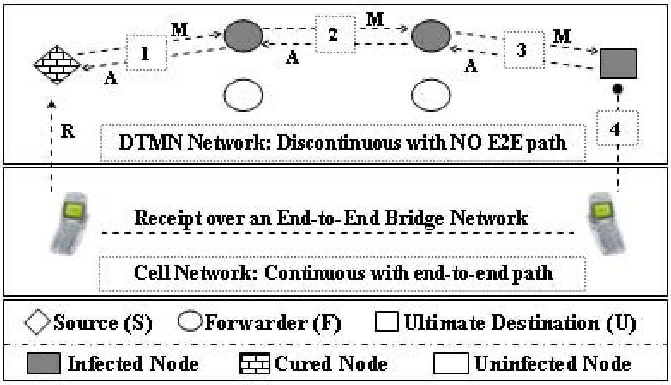

M
M
M
1
2
3
4
A
A
A
R
DTMN Network: Discontinuous with NO E2E path
Receipt over an End-to-End Bridge Network
Cell Network: Continuous with end-to-end path
Source (S)
Forwarder (F)
Ultimate Destination (U)
Infected Node
Cured Node
Uninfected Node

Fig. 4. The network-bridged receipt reliability approach

then be transmitted to these nodes. The result is a very rapid cure for all infected nodes in the network.

The advantage of the network-bridged approach is to reduce the round trip time between nodes S and U roughly by half. Consequently, the message is dropped faster from the queue in A since the receipt arrives faster. The drawback, however, lies in the assumption itself: the added complexity of bridging the DTMN network with the cell network. We believe, however, that the interconnection of these two networks is a likely possibility in the future.

5 Evaluation

The primary goal of our evaluation is to compare the performance and examine the tradeoffs between the reliability approaches described in Section 4. We first describe our simulation setup and environment. We then summarize the outcomes of an extended set of simulations we conducted. The extended result set is not shown due to space limitation. Therefore, we only present a subset of our results that most clearly allows us to show the tradeoffs between our reliability approaches.

5.1 Simulation Environment

We conducted our simulations using the GloMoSim network simulator. We added an overlay layer that handles message bundle relaying and implements the reliability approaches that we have described. We use a *modified* random way-point mobility model that avoids the major problem of node slow down in the conventional random way-point model. We believe this model closely approximates the scenarios with which we are concerned, such as battlefields or disaster rescue operations, due to their hostility and unpredictable movement. The node speed ranges between 20 to 35 meters per second, and the rest period is between 0 and 10 seconds. We examined other ranges as well, and they produced similar results with respect to our reliability approaches. Every point in our results is taken as an average of ten different seeds.

Table 1. Simulation Parameters

Parameter	Value Range	Nominal Value
Terrain	$10km^2$ to $50km^2$	$10km^2$
Number of Nodes	10 to 250	100
Simulated Time	1hour to 24 hours	6 hours
Beacon Interval	0.5sec to 50sec	1sec
Times-To-Send	1 to 50	10
Retransmission Wait Time	0sec to 500sec	50sec
Reliability Approach	Hop-by-hop, Active, Passive or Network-Bridged	N/A

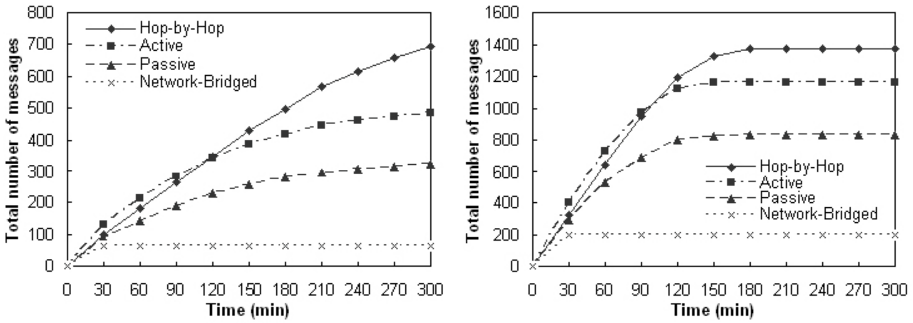
The major parameters used in our simulations are summarized in Table 1. The *Terrain* is the area over which the *Number of Nodes* are scattered. *Simulated Time* represents the amount of time the simulations run. The *Beacon Interval* is the period after which beacons are sent. A "beacon" is simply a signal emitted by all nodes to search for other nodes in the network as well as to announce its location. The *Times-To-Send* (TTS) is the number of times a node will successfully forward a message to other nodes in the network. *Retransmission Wait Time* represents the amount of time a node remains idle after successfully forwarding a message to another node. When the retransmission wait time expires, the node then tries to resend the same message. We mainly use TTS to represent the *willingness* of the nodes to participate in message relaying. Finally, the *reliability approach* parameter represents our four different acknowledgement schemes.

We consider three main metrics in evaluating our reliability approaches. The first metric is *Cost*, which is the total number of messages sent by all nodes in the network. The second metric is *Queuing Time*, which is the average time a message remains in the sender node's queue before it is dropped. The third metric we consider is *Delivery Ratio*, which is the percentage of messages delivered. We choose to focus on the first two metrics since delivery ratios in DTMNs simply depend on the time ceiling set for message delivery, i.e. given enough time, all messages will eventually be delivered.

5.2 Results

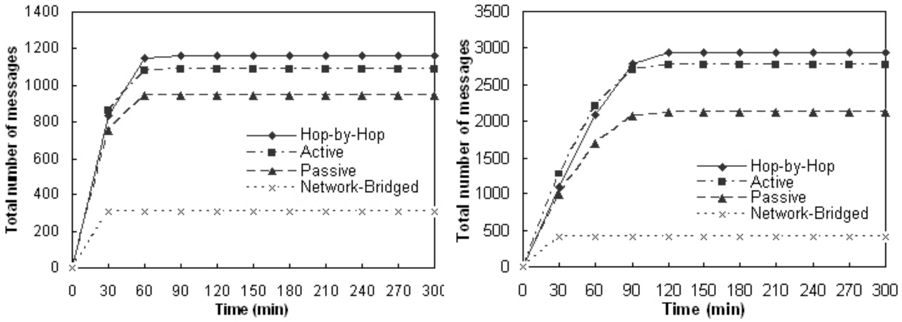
We present a summary of the extended set of simulations, along with a subset of our simulation results, which clarify and support our conclusions. All the results are shown for a single sender node sending one message to a single ultimate destination. The purpose of our simulations is twofold. First, we hope to better understand how different reliability approaches behave when run in a DTMN. Second, we want to understand the tradeoffs between these approaches.

Generally speaking, the network-bridged receipt incurs the least cost when compared to the other approaches. The highest cost, on the other hand, occurs with the hop-by-hop approach. The cost of the active and passive receipts fall in between, with active receipt being relatively more expensive. These observations



(a) 25 nodes (b) 50 nodes

Fig. 5. The cost of the reliability approaches over time in DTMNs with different node densities. Graphs (a) and (b) represent 25 and 50 nodes, both with a TTS of 10.

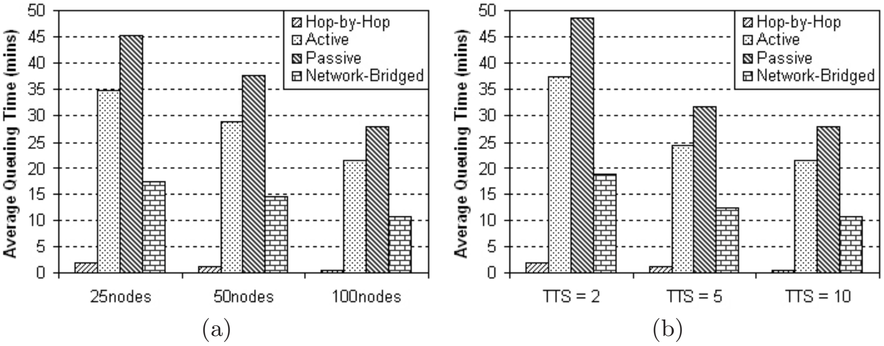


(a) Times to send = 5 (b) Times to send = 10

Fig. 6. The cost of the reliability approaches over time in DTMNs with different willingness levels. Graphs (a) and (b) represent TTS of 5 and 10, both with 100 nodes.

are supported by Figure 5 and Figure 6, which demonstrates the cost of each reliability approach in terms of the total number of messages sent. We measure this cost under different network densities, 25 nodes in Figure 5(a), 50 nodes in Figure 5(b) and 100 nodes in Figure 6(d), as well as different willingness levels, times-to-send is set to 5 in Figure 6(a) and 10 in Figure 6(b). One interesting observation is where the cost of the active receipt is the highest until it is eventually exceeded by the hop-by-hop approach. This result is because after the message reaches the ultimate destination, we now have two messages infecting the network, which creates this large cost. Eventually, however, the receipt cures those nodes infected with the original message and is itself cured after reaching the source node. We note also that changes in node density or willingness levels have minor impact on the *relative* performance of our reliability approaches.

Even though the performance of the reliability approaches is relatively similar over different network densities, other aspects, such as the rate of message spreading and convergence, vary. This result is particularly evident in the



(a) (b)

Fig. 7. The impact of (a) the number of nodes, and (b) the times-to-send on the average queuing time of a message at the sender node

difference in the Y-axis scales of Figure 5 and Figure 6. Generally speaking, the messages spread faster in denser networks. This observation can be seen by the sharper increase in the total number of messages in the case of Figure 6(b) when compared to Figures 5(a) and Figure 5(b). We compare Figure 6(b) with Figure 5 since the former measures the cost over a 100 node network with the same TTS value of 10 as that used in Figure 5. Alternatively, the network heals faster in denser networks. This result is shown in the faster convergence of the lines in Figure 5(b) when compared to those in Figure 5(a). This convergence leads to a steady horizontal line because the network reaches a point of saturation where it no longer needs to forward the message.

Regarding the average queuing time, the results show that the hop-by-hop approach has the lowest value. This low value is because the source node does not wait for any end-to-end acknowledgement to be received, and therefore, drops the message from its buffer after forwarding to other nodes in the network. If end-to-end reliability is required, the best approach in terms of minimal queuing time is the network-bridged approach. Figure 7 supports these observations by illustrating the average queuing time of a given message with respect to our reliability approaches under (a) different densities, and (b) different willingness levels. The other interesting fact Figure 7 highlights, is that active receipt has less queuing time than passive receipt. This fact offers a tradeoff for the extra cost incurred in the active receipt when compared to the passive receipt approach. The reason for this result is due to the active way in which the receipt is sent when compared to the passive approach. The active approach results in the receipt reaching the source faster, but at a higher cost.

Figure 7 also shows that the tradeoffs between the reliability approaches is generally similar over different densities and different willingness levels. The primary difference is that the overall queuing time of all the reliability approaches decreases as the network density or willingness levels increase. This result is because in denser networks, or when nodes are trying harder to forward a message, the overall end-to-end delay decreases. This decrease in delay consequently leads to smaller queuing time.

6 Conclusions and Future Work

In this paper, we have considered transport layer issues, specifically reliability, over a special class of DTNs knows as DTMNs. We introduced four different reliability approaches: hop-by-hop, active receipt, passive receipt, and network-bridged receipt. We have investigated and evaluated these approaches via simulation. Overall, we discovered that the choice of the most suitable reliability approach depends on the expected complexity of the underlying DTMN. For example, the hop-by-hop is the simplest, while network-bridged is the most complex. Also, the priority of cost versus delay governs the choice between the active and passive receipt.

We consider this paper a next step in thoroughly investigating transport layer issues in DTNs in general. Our future work, therefore, is to apply these approaches to DTNs in general, and see how they might be modified and applied to other DTN architectures. Also, we intend to address other transport layer issues, particularly, congestion control.

References

1. University of South Florida: Center for robot-assisted search and rescue. http://crasar.csee.usf.edu/.
2. V. Cerf, et. al. Interplanetary Internet (IPN): Architectural Definition. *IETF Internet Draft, draft-irtf-ipnrg-arch-00.txt*, May 2001.
3. DTNRG. Delay Tolerant Networking Research Group. http://www.dtnrg.org/.
4. K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *ACM SIGCOMM*, Karlsruhe, Germany, August 2003.
5. K. Fall, W. Hong, and S. Madden. Custody Transfer for Reliable Delivery in Delay Tolerant Networks. *Intel Research, Berkeley-TR-03-030*, July 2003.
6. K. Harras, K. Almeroth, and E. Belding-Royer. Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks. In *IFIP Networking*, Waterloo, Canada, May 2005.
7. A. Hooke. The Interplanetary Internet. *Communications of the ACM*, 44(9):38–40, September 2001.
8. S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *ACM SIGCOMM*, Portland, OR, August 2004.
9. D. Johnson and D. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*, volume 353. Kluwer Academic Publishers, 1996.
10. P. Juang, et. al. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences With ZebraNet. In *In International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, October 2002.
11. E. Krotkov and J. Blitch. The Defense Advanced Research Projects Agency (DARPA) Tactical Mobile Robotics Program. *The International Journal of Robotics Research*, 18(7):769–776, July 1999.
12. Q. Li and D. Rus. Sending Messages to Mobile Users in Disconnected Ad-Hoc Wireless Networks. In *ACM MobiCom*, pages 44–55, Boston, MA, August 2000.
13. A. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.

14. C. Perkins. Ad-hoc On-Demand Distance Vector Routing. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.
15. C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM*, pages 234–244, London, England, October 1994.
16. E. Royer and C. Toh. A Review of Current Routing Protocols for Ad-hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, 6(2):46–55, April 1999.
17. R. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks. In *In IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, 2003.
18. A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. *Technical Report CS-200006, Duke University*, April 2000.
19. W. Zhao and M. Ammar and E. Zegura. Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network. In *IEEE INFOCOM*, Miami, FL, March 2005.
20. W. Zhao, M. Ammar, and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *ACM MobiHoc*, Tokyo, Japan, May 2004.

Performance of Competing High-Speed TCP Flows

Michele C. Weigle, Pankaj Sharma, and Jesse R. Freeman IV

Department of Computer Science, Clemson University, Clemson, SC 29634
{mweigle, pankajs, jessef}@cs.clemson.edu

Abstract. The goal of recent high-speed TCP implementations is to allow scientists who have access to new high-speed networks to efficiently transfer large datasets to their remote colleagues. As of yet, there is no standard high-speed TCP. Because of this, scientists using one high-speed protocol may find themselves sharing a link with scientists using a different high-speed protocol. Previous work has evaluated such inter-protocol performance, but only with both flows starting at the same time – an unlikely situation. We perform an evaluation study using *ns*-2 to investigate the performance of competing high-speed TCP flows where one flow enters a network in which another high-speed flow has already reached its maximum data rate. The fairest result would be for the existing flow to cede half of its bandwidth to the new flow in order to allow both flows to evenly share the link. Our results show that in most cases this does not happen, but rather one high-speed flow dominates the other. Surprisingly, it is not always the existing flow that dominates.

Keywords: High-speed TCP, congestion control, performance evaluation, network simulation.

1 Introduction

Recently, several new variants of TCP have been developed to take advantage of high capacity networks. It has been shown that Standard TCP, which handles most Internet traffic, has limitations when a single connection attempts to send data at very high speeds (*i.e.*, faster than 100 megabits per second) over long distances [6]. These new high-speed variants of TCP were designed to solve the limitations with high-speed transfers while maintaining reliability and fairness to Standard TCP flows. The most prominent of these are HighSpeed TCP (HS-TCP) [6, 7], Scalable TCP (S-TCP) [10], FAST TCP [8, 9], H-TCP [15], BIC-TCP [18], and CUBIC [13].

The target users of high-speed protocols are scientists who have access to fast, long-distance links that connect them to their colleagues in other locations. Distributed, collaborative applications for analyzing large data sets require a reliable and fast mechanism for distributing the data. Since the use of a dedicated high-speed link from one lab to another is prohibitively expensive, it is more likely that a network of connected research labs, much like the National LambdaRail project [1], will be developed. Scientists cannot rely on a single high-speed flow being allowed to consume the entire capacity of a link. The high-speed flow will likely have to share the capacity with other high-speed flows, as well as flows from low-speed applications such as web, email, and file sharing. Further, it may not be the case that a single high-speed technology is

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 476–487, 2006.

adopted by the entire community, but that several of these TCP variants will co-exist on the same links. For this reason, these new high-speed TCP implementations should be tested in an environment as close as possible to their likely real-world deployment.

Most previous work (including [2, 3, 16, 17]) has investigated either how these high-speed TCP implementations perform individually on dedicated links or how fairly they share link bandwidth with Standard TCP. Bullot *et al.* [3] determined that most of these protocols compete rather fairly against each other when two flows using different protocols share a single link, but they only tested the case where both flows started at the same time. Unlike Bullot *et al.*, we assume that one flow is sending at a high data rate before another high-speed flow enters the network. The ideal outcome should be for the existing flow to cede half of its bandwidth to the new flow so that both can fairly share the link.

We ran sets of experiments in the *ns*-2 network simulator [12] where we tested two competing high-speed TCP flows in the situation where one flow started well before the other flow. We studied all combinations of HS-TCP, S-TCP, FAST, H-TCP, BIC-TCP, and CUBIC in a network with a 622 Mbps (OC-12) bottleneck and a 100 ms RTT. We ran experiments with the maximum router queue buffer length at 100% of the bandwidth-delay product (BDP), 20% of the BDP, and 40 packets. We consider this to be the first step in a larger study of high-speed TCP protocols that also investigates the impact of other parameters such as RTTs, background traffic, reverse path traffic, and queuing mechanism.

We find that a 20% BDP router queue buffer results in high link utilization for these flows, intra-protocol fairness suffers when competing flows are started at different times, S-TCP is too aggressive in obtaining throughput from other high-speed flows, and in general, most of the high-speed protocols are not fair when competing with other high-speed protocols.

2 Background

All of the high-speed protocols that we evaluate attempt to be fair to Standard TCP flows that might be sharing the link. These protocols use Standard TCP when the TCP window w is less than a threshold value, and only use the high-speed version when w is above the threshold. Here we present a very brief overview of each of the protocols.

HS-TCP: When an acknowledgment (ACK) is received, HS-TCP increases w by $a(w)/w$. When one or more losses is detected in an RTT, HS-TCP sets w to $(1-b(w))w$. The goal is for a more aggressive increase and less aggressive decrease than Standard TCP in low-loss environments (*i.e*, environments where w is allowed to grow past the threshold, *LowWindow*). Current implementations of HS-TCP use a lookup table to determine the values of $a(w)$ and $b(w)$. Recommended settings allow $a(w)$ in the range of [1, 72] segments and $b(w)$ in the range [0.1, 0.5].

S-TCP: S-TCP is a simplification of HS-TCP, where the window adjustment functions $a(w)$ and $b(w)$ no longer depend on w. When the congestion window w is greater than *LowWindow*, S-TCP sets a to $0.01w$ (so that w increases by 0.01 for each returning ACK) and b to 0.125. Like HS-TCP, when w is less than *LowWindow*, S-TCP behaves like Standard TCP.

FAST TCP: FAST TCP is a delay-based protocol that uses increasing RTTs as a form of congestion notification. In FAST, the congestion window w is updated every other RTT based on a function of the observed RTT and α, the number of segments that FAST attempts to keep in the network. If the RTT increases, FAST may decrease w even though loss has not occurred. Upon loss, FAST TCP behaves the same as Standard TCP (*i.e.*, it reduces w by half and enters TCP loss recovery).

H-TCP: H-TCP increases w based on Δ, the time between successive congestion events, and β, the ratio of the minimum-observed RTT to the maximum-observed RTT. H-TCP uses Δ^L (default of 1 second) as a threshold for entering high-speed mode. Upon detection of packet loss, $w = \beta w$, where $0.5 \leq \beta \leq 0.8$. These settings allow H-TCP to be *RTT-fair*, meaning that flows with longer RTTs will see similar throughput to flows with shorter RTTs. In addition, H-TCP flows in high-speed mode (where $\Delta > \Delta^L$) will cede some of their throughput to newer flows that have not yet reached high-speed mode.

BIC-TCP: Like H-TCP, BIC-TCP strives to maintain RTT-fairness. BIC-TCP sets a minimum window size w_{min}, maximum window size w_{max}, and a target window size w_{target}, which is the midpoint between w_{min} and w_{max}. BIC-TCP uses a binary search algorithm to reach the target window size (with a maximum increment of S_{max} segments in one step). When loss occurs, w_{max} is set to the current window size w, and w_{min} is set to the reduced window size $(1-\beta)w$, where $\beta = 0.125$. A new target w_{target} is then computed (as the midpoint between w_{min} and w_{max}). When the congestion window reaches the target without experiencing loss, the current window size becomes the new minimum ($w_{min} = w$) and a new target is computed.

CUBIC: CUBIC is a modification of BIC-TCP with the goal of improving on BIC-TCP's fairness. In CUBIC, the window increase is determined by a cubic function $w = C(t-K)^3 + W_{max}$, where C is a constant used for scaling, t is the time since the window was last reduced, W_{max} is the size of the window just before the window was last reduced, and $K = \sqrt[3]{W_{max}\beta/C}$, where β is a constant decrease factor. When a loss occurs, the window is reduced to $W_{max}\beta$, where $\beta = 0.8$.

3 Methodology

We ran all experiments in the *ns*-2 network simulator using the topology shown in Figure 1. Two senders are on the left side of the network, and two receivers are on the right side of the network. Each end node is connected to a router by a 1 Gbps link with a propagation delay of 1 ms. The two routers are connected by a 622 Mbps bottleneck link with a 48 ms propagation delay. This topology gives each sender a 100 ms round-trip time (RTT). The network has a bandwidth-delay product (BDP) of 7775 1000-byte segments, and drop-tail queuing is used at both routers. We performed the full set of experiments with three different router queue buffer lengths: 100% of the BDP, 20% of the BDP, and 40 segments. To ensure that TCP window size is not a limiting factor, each TCP connection has a maximum window size of 67,000 segments, which is about 64 MB. In each simulation, two connections are started, one from node 0 and one from

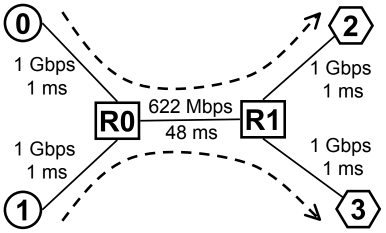


Fig. 1. Network Topology

node 1. The first connection (flow 1) is started at time 0, and the second connection (flow 2) is started 50 seconds later. Each simulation is run for 500 seconds.

Even though using the same RTT for both flows could cause synchronized loss, we were more concerned with factoring out the effects of RTTs on our results. Previous work [18] has shown that many of the high-speed protocols are not *RTT-fair*, meaning that flows with shorter RTTs achieve higher throughput than flows with longer RTTs. In order to concentrate on the effects of competing flows, we chose to equalize the RTTs.

We tested six high-speed protocols (HS-TCP, S-TCP, FAST, H-TCP, BIC-TCP, and CUBIC) by running a set of six experiments for each protocol and maximum router queue buffer size. Within each set, flow 1 uses the same protocol, and flow 2 uses a different one of the six protocols. For example, in the HS-TCP set, flow 1 is always HS-TCP and flow 2 is either HS-TCP, S-TCP, FAST, H-TCP, BIC-TCP, or CUBIC.

For each protocol, we used the parameters recommended by the protocol's authors. More details of the experimental setup we used (including *ns*-2 scripts) can be found at `http://www.cs.clemson.edu/~mweigle/research/hstcp/`.

We use the asymmetry metric from Bullot *et al.* [3] to evaluate the fairness of the various proposals. This metric, as opposed to the Chiu and Jain fairness index [4], provides information about which flow is more aggressive rather than just if the flows share the link fairly. The asymmetry metric is defined as $A = (\bar{x}_1 - \bar{x}_2)/(\bar{x}_1 + \bar{x}_2)$, where $\bar{x}_i$ is the average throughput obtained for flow i. Average throughput was measured starting at time 250 seconds to focus on steady-state throughput. The closer the asymmetry metric A is to 0, the more fair the distribution of throughput. The closer A is to 1, the more flow 1 dominates the transfer, and the closer A is to -1, the more flow 2 dominates.

4 Results

We found that with a router queue buffer length of 40 packets, utilization suffered, with many pairs of flows together obtaining less than 50% of the total link capacity. A queue buffer length of 100% BDP provides the best link utilization, but may be an unrealistic size for real networks. In our network, 100% BDP is 7775 packets, which is larger than the maximum queue size on many commercial routers.[1] With a 20% BDP queue buffer length, the maximum queue size was 1555 packets (in the range of commercial routers) and all experiments had a total link utilization of 99%-100%. With regard to fairness,

[1] For example, Cisco routers typically have a default output queue size of 40 packets, with a maximum size of 4096 packets.

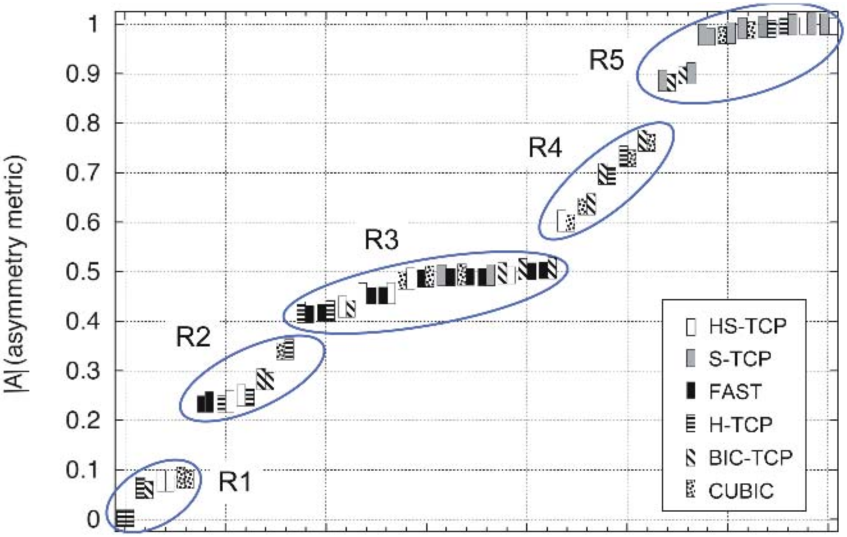


Fig. 2. We show the absolute value of the asymmetry metric. Each experiment is represented by two rectangles, in which the pattern on the left indicates flow 1's protocol and the pattern on the right indicates flow 2's protocol. The taller rectangle indicates the flow that received the larger share of the throughput. We divide the experiments into 5 regions based on fairness.

the results obtained with the 20% BDP queue buffer length were slightly fairer in many cases than with either 100% BDP or 40 packets. For the remainder of the paper, we focus on results obtained with a queue buffer length of 20% BDP. We sorted the results of the experiments into five regions, based on the absolute value of *A* for the experiment (Figure 2). We will discuss the experiments that fell into each region separately. In general for all experiments, the behavior of the flows reached some steady state before the simulation ended. For each region, we will show graphs of the congestion window (in segments) for two representative experiments.

4.1 Region 1

In Figure 3, we show the congestion windows of two representative experiments from Region 1, which represents the fairest of the protocol pairings. Three out of the four pairings are intra-protocol (H-TCP, HS-TCP, and CUBIC), which is not surprising. Since both flows are running the same AIMD window adjustment algorithm, they should eventually converge to a fair share no matter when the individual flows are started [4]. For the intra-protocol pairings that fall into Region 1, the two H-TCP flows are the fastest to converge - around time 150 seconds. Since we do not measure throughput for computing *A* until time 250, the H-TCP intra-protocol pairing has an *A* value of 0.0. The CUBIC intra-protocol flows do not converge until about 450 seconds, and the two HS-TCP flows do not converge until about 500 seconds.

The pairing of H-TCP and BIC-TCP when H-TCP starts first is the only inter-protocol experiment in Region 1. When the order of protocols is reversed though, the behavior is much less fair (falls into Region 4), with BIC-TCP controlling most of the

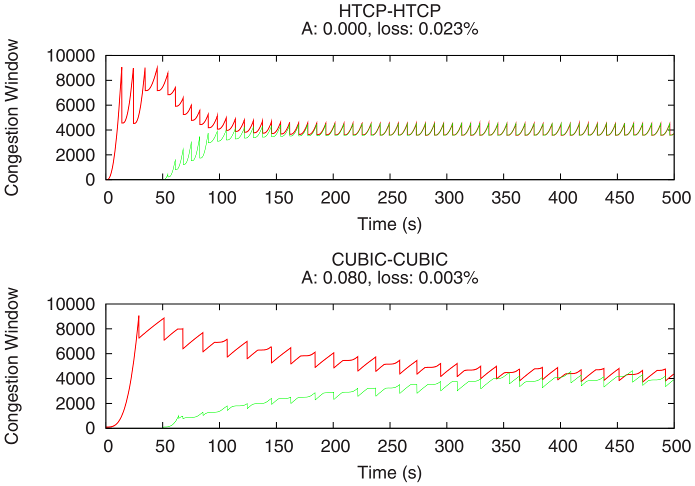


Fig. 3. Region 1: We show the congestion window for the H-TCP intra-protocol and CUBIC intra-protocol experiments

throughput. H-TCP is able to cede some of the bandwidth when the BIC-TCP flow enters late, but when the roles are reversed, BIC-TCP does not cede a fair share of the bandwidth to the entering H-TCP flow. The behavior of both BIC-TCP and H-TCP is dependent upon competing traffic, especially if that traffic is causing the queue to overflow. Both of these protocols include a *time since last loss* factor in their window adjustment policies (BIC-TCP implicitly and H-TCP explicitly). These protocols will increase their aggressiveness the longer they go without experiencing a loss. This is in contrast to protocols like HS-TCP or S-TCP whose aggressiveness only depends on the current window size.

4.2 Region 2

Figure 4 shows the congestion windows for two representative pairings from Region 2. Both the FAST and BIC-TCP intra-protocol experiments fell into this region. Neither of these pairings converged by time 500, which is why these did not fall into Region 1 instead. All of the inter-protocol experiments in Region 2 involved H-TCP, including both pairings of H-TCP and HS-TCP. In these experiments, HS-TCP obtained more throughput than H-TCP even when HS-TCP started later. With CUBIC and H-TCP, even though H-TCP started later, it gained higher throughput than CUBIC.

For the FAST intra-protocol experiments, when the flows start at different times, flow 1 occupies its share of the queue (according to α) and keeps the queue stable. When the second flow enters, its estimate of the minimum RTT is inaccurate because it sees the queuing delay caused by flow 1's packets as the minimum. With this underestimate of the actual queuing delay, flow 2 takes more than its fair share of the network resources. This problem does not occur when competing with other types of protocols because the other protocols drive the queue to overflow and drain completely.

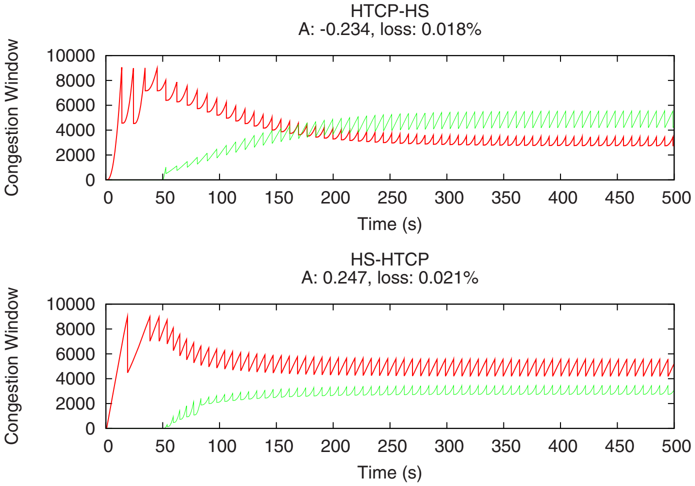


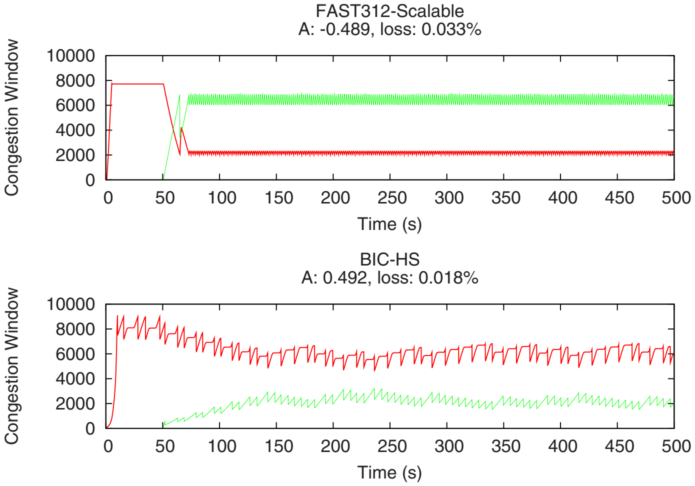
Fig. 4. Region 2: We show the congestion window for both pairings of H-TCP and HS-TCP. The top graph shows when HS-TCP is flow 1, and the bottom graph shows when H-TCP is flow 1.

4.3 Region 3

In Figure 5, we show the congestion windows of two representative experiments from Region 3. The experiments in this region all produced very similar results. Ten of the thirteen experiments involved FAST. Since FAST is a delay-based protocol, as the queuing delay increases, FAST adjusts its window either by increasing more slowly than before, or by decreasing, depending on the degree of the increase in the queuing delay. When competing against loss-based protocols that have to fill the queue to determine when available bandwidth is exhausted, FAST will see poorer performance than when competing against flows that are also sensitive to changes in the queuing delay (*i.e.*, another FAST flow). When we look at the queue size in experiments where the FAST flow starts first, the queue size is very stable – the queue neither fills nor drains. (This behavior, though is very dependent upon the size of the queue buffer and on FAST's α parameter.) Once the second flow enters, the queue becomes bursty. For all of these experiments, the FAST flow sees lower throughput than the other protocols. This is more a characteristic of how FAST competes against loss-based protocols than the aggressiveness of the other protocols.

One interesting point about FAST is its competition with S-TCP. The experiments in which S-TCP competed with FAST fall into Region 3. All other experiments involving S-TCP fall into Region 5, where S-TCP is always the more aggressive flow. FAST keeps its share of packets in the queue even as S-TCP drives the queue to overflow. Additionally, since FAST backs off as the queuing delay increases, it is able to avoid much of the loss caused by S-TCP overflowing the queue. Once both flows are in steady-state, the queue never completely drains, resulting in very high link utilization.

Also in Region 3 are both pairings of BIC-TCP and HS-TCP. For these, flow 1 sees higher throughput regardless of the protocol. When competing against each other, both



FAST312-Scalable
A: -0.489, loss: 0.033%

BIC-HS
A: 0.492, loss: 0.018%

Fig. 5. Region 3: We show the congestion window for the pairings of FAST and S-TCP and of BIC-TCP and HS-TCP

of these protocols are too aggressive in keeping bandwidth already obtained and not aggressive enough in grabbing its share of bandwidth from the existing flow.

4.4 Region 4

In Figure 6, we show the congestion windows of two representative experiments from Region 4. All but one of the pairings in Region 4 contain a CUBIC flow, and in all of these CUBIC gets less throughput than its competitor. Over all of the experiments, the only time that CUBIC saw higher throughput than its competitor was against FAST, but we have already mentioned that this is more due to FAST's behavior than CUBIC's. The designers of CUBIC consciously made the protocol behave less aggressively than BIC-TCP so that it would be fairer to competing flows. We see that two CUBIC flows share more fairly than two BIC-TCP flows, but CUBIC is not aggressive enough when competing against other protocols. One reason for the behavior we see might be due to the synchronized loss patterns that we get in a study such as ours without background traffic. Like BIC-TCP and H-TCP, CUBIC has an window increase function that depends upon the time elapsed between successive congestion events. If CUBIC can avoid some losses when the queue overflows (*i.e.*, the other flows see loss, but not CUBIC), then the CUBIC flow will be able to gain some of the available bandwidth released when the other flow backs off.

4.5 Region 5

All of the experiments in Region 5 contain at least one S-TCP flow, and the S-TCP flow always sees much higher throughput than the other flow. In Figure 7, we show how aggressive S-TCP is in obtaining and keeping bandwidth. The top graph shows S-TCP when competing against a BIC-TCP flow that is using the entire link before the

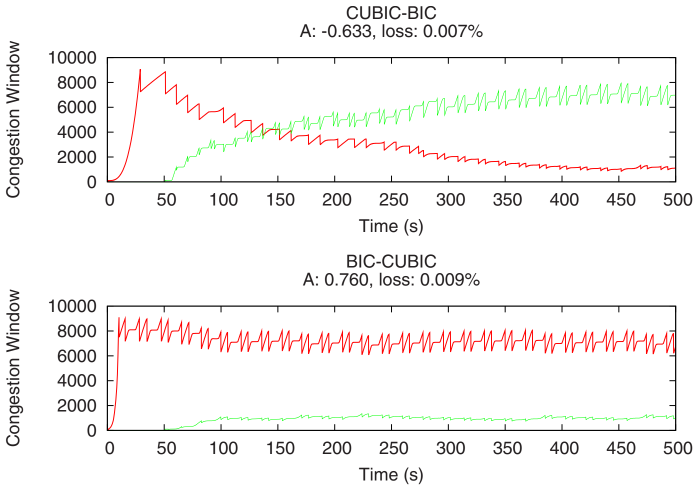


Fig. 6. Region 4: We show the congestion window for both pairings of CUBIC and BIC-TCP. The top graph shows CUBIC as flow 1, and the bottom graph shows when BIC-TCP is flow 1.

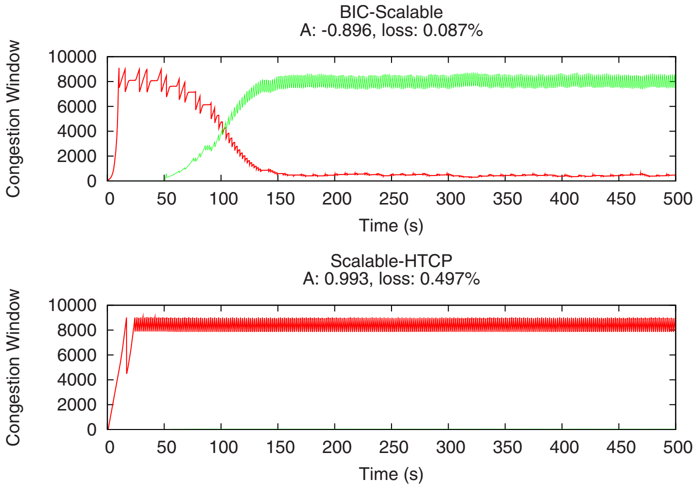


Fig. 7. Region 5: We show the congestion window for the pairings of BIC-TCP and S-TCP and of S-TCP and H-TCP. Note that on the bottom graph, the line for HTCP is not visible as its congestion window was very close to 0.

S-TCP flow begins. Relatively quickly, S-TCP increases its window and pushes BIC-TCP down to very little throughput. In the bottom graph, S-TCP is competing against an H-TCP flow that enters late. The congestion window of the H-TCP flow is not visible on the graph because the S-TCP flow does not back off long enough after loss to allow the H-TCP flow to take advantage of the newly available bandwidth. S-TCP in essence is a MIMD (multiplicative increase, multiplicative decrease) protocol instead of the standard AIMD. Chiu and Jain [4] have shown that MIMD protocols do not converge to fairness. With S-TCP, the higher the window, the larger the amount of increase. So, when S-TCP does encounter loss, it is able to increase its window very quickly to take back the available bandwidth it had given up.

4.6 Flows Starting at the Same Time

We also ran experiments where we started both flow 1 and flow 2 at the same time. We found that intra-protocol results improved for all cases, with convergence times essentially going to 0. The improvement was most dramatic for S-TCP, shown in Figure 8. In the top graph, we see the same behavior as when S-TCP was competing with H-TCP in Figure 7. When the two S-TCP flows start at the same time (and since they have the same RTT), the windows match and the competition is fair. Once one of the S-TCP flows gains an advantage, it will keep increasing its advantage due to its MIMD window adjustment algorithm.

FAST also improves its intra-protocol fairness performance when both flows start at the same time. When discussing the intra-protocol FAST results in Region 2, we mentioned that the later-joining, second FAST flow has an inaccurate estimate of the minimum RTT. When both FAST flows start at the same time, they both have the same estimate of the minimum RTT (and thus, the queuing delay).

We found that flow start-time also had an effect on BIC-TCP. When BIC-TCP and either HS-TCP or H-TCP started at the same time, BIC-TCP increased its window aggressively and did not let the other flows share fairly. The behavior was very similar to results obtained when the BIC-TCP flow started first and either an HS-TCP or an H-TCP flow joined later.

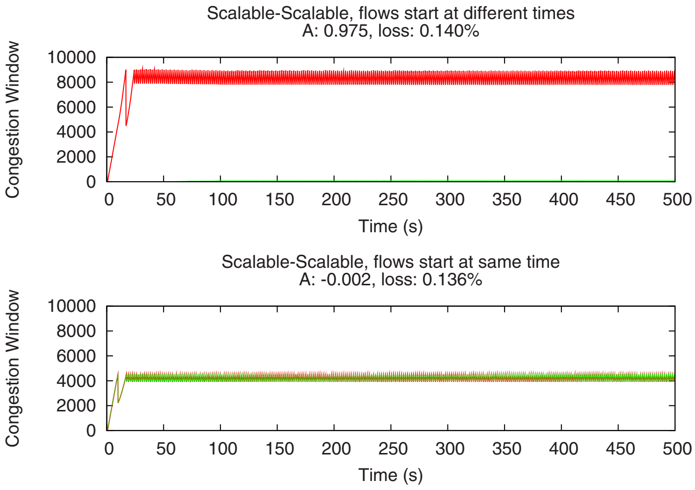


Fig. 8. We show the congestion window for two intra-protocol experiments with S-TCP. The top graph is when flow 1 starts 50 seconds before flow 2, and the bottom graph shows when both flows start at the same time. Note that on the top graph, the line for flow 2 is not visible as its congestion window was very close to 0.

For all of the other pairings, the start time did not seriously impact fairness. In general, protocols that were so unfair that they overtook the flow that started first were still unfair when the flows started at the same time.

5 Conclusions and Future Work

We studied the performance of two competing TCP flows in the scenario where one flow enters the network after the other flow has fully utilized the link. We evaluated the performance of six high-speed protocols using three different router queue buffer lengths. We concentrated on the results obtained with a queue buffer length of 20% BDP, which provided high link utilization and a realistic buffer size. We make the following findings:

- In general, most of the high-speed protocols are not fair when competing with other high-speed protocols.
- Intra-protocol fairness suffers when the flows are started at different times, due to slower convergence times.
- The performance of S-TCP and FAST do not depend upon the competing flow, but rather are dependent only upon their own operation.
- S-TCP is too aggressive in obtaining bandwidth, even when competing with another S-TCP flow.

As part of our future work, we would like to add background HTTP traffic and reverse path traffic to simulate typical non-high-speed Internet traffic that may be sharing the link with the high-speed flows. Also, we plan to study how using active queue management (AQM) techniques in these more realistic environments might affect the high-speed protocols. For this work, we plan to build upon previous studies of AQM and high-speed protocols [16, 17, 18].

Acknowledgements

We thank Injong Rhee and Lisong Xu for the *ns*-2 source code for S-TCP, BIC-TCP, and CUBIC and for the use of simulation scripts [14]. We thank Tony Cui and Lachlan Andrew from the University of Melbourne's Centre for Ultra-Broadband Information Networks (CUBIN) for *ns*-2 source code for FAST [5], and we thank Douglas Leith *et al.* for the *ns*-2 source code for H-TCP [11].

References

[1] National LambdaRail Project. `http://www.nlr.net/`.

[2] A. Antony, J. Blom, C. de Laat, J. Lee, and W. Sjouw. Microscopic examination of TCP flows over transatlantic links. *iGrid 2002 Special Issue, Future Generation Systems*, 19(6), 2003.

[3] H. Bullot, R. L. Cottrell, and R. Hughes-Jones. Evaluation of advanced TCP stacks on fast long-distance production networks. *Journal of Grid Computing*, 1(4):345–359, 2003. Graphs available at `http://www-iepm.slac.stanford.edu/bw/tcp-eval/`.

[4] D. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, pages 1–14, June 1989.

[5] T. Cui and L. Andrew. FAST TCP simulator module for ns-2, version 1.1, 2004. Available at `http://www.cubinlab.ee.mu.oz.au/ns2fasttcp`.

[6] S. Floyd. Highspeed TCP for large congestion windows, Dec. 2003. RFC 3649, Experimental.

[7] S. Floyd, S. Ratnasamy, and S. Shenker. Modifying TCP's congestion control for high speeds, May 2002. Technical Note, available at `http://www.icir.org/floyd/papers/hstcp.pdf`.

[8] C. Jin, D. X. Wei, and S. H. Low. FAST TCP: motivation, architecture, algorithms, performance. In *Proceedings of IEEE INFOCOM*, Hong Kong, Mar. 2004.

[9] C. Jin, D. X. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot, and S. Singh. FAST TCP: From theory to experiments. *IEEE Network*, 19(1):4–11, January/February 2005.

[10] T. Kelly. Scalable TCP: Improving performance in highspeed wide area networks. In *Proceedings of PFLDnet*, Geneva, Switzerland, Feb. 2003.

[11] D. Leigh, R. Shorten, et al. H-TCP ns-2 implementation, 2005. Available at `http://www.hamilton.ie/net/research.htm#software`.

[12] S. McCanne and S. Floyd. ns Network Simulator. Software available at `http://www.isi.edu/nsnam/ns/`.

[13] I. Rhee and L. Xu. CUBIC: A new TCP-friendly high-speed TCP variant. In *Proceedings of PFLDnet*, Lyon, France, Feb. 2005.

[14] I. Rhee and L. Xu. Simulation code and scripts for CUBIC, 2005. Available at `http://www.csc.ncsu.edu/faculty/rhee/ export/bitcp/cubic-script/script.htm`.

[15] R. N. Shorten and D. J. Leith. H-TCP: TCP for high-speed and long-distance networks. In *Proceedings of PFLDnet*, Argonne, Illinois, Feb. 2004.

[16] E. Souza and D. A. Agarwal. A HighSpeed TCP study: Characteristics and deployment issues. Technical Report LBNL-53215, 2003. Available at `http://dsd.lbl.gov/~evandro/hstcp/hstcp-lbnl-53215.pdf`

[17] K. Tokuda, G. Hasegawa, and M. Murata. Performance analysis of HighSpeed TCP and its improvements for high throughput and fairness against TCP Reno connections. In *High-Speed Networking Workshop (HSN)*, 2003.

[18] L. Xu, K. Harfoush, and I. Rhee. Binary increase congestion control for fast, long distance networks. In *Proceedings of IEEE INFOCOM*, Hong Kong, Mar. 2004.

On the Accuracy of Analytical Models of TCP Throughput

Ibtissam El Khayat, Pierre Geurts, and Guy Leduc

Department of Electrical Engineering and Computer Science,
University of Liège
{elkhayat, geurts, leduc}@montefiore.ulg.ac.be

Abstract. Based on a large set of TCP sessions we first study the accuracy of two well-known analytical models (SQRT and PFTK) of the TCP average rate. This study shows that these models are far from being accurate on average. Actually, our simulations show that 70% of their predictions exceed the boundaries of TCP-Friendliness, thus questioning their use in the design of new TCP-Friendly transport protocols. Our study also shows that the inaccuracy of the PFTK model is largely due to its inability to make the distinction between the two packet loss detection methods used by TCP: triple duplicate acknowledgments or timeout expirations. We then use supervised learning techniques to infer models of the TCP rate. These models show important accuracy improvements when they take into account the two types of losses. This suggests that analytical model of TCP throughput should certainly benefit from the incorporation of the timeout loss rate.

1 Introduction and Motivation

TCP is a transport protocol widely used by applications like remote access (ssh, telnet), file transfer (ftp), and Peer-to-Peer. It occupies more than 90% of Internet resources [8]. The success of this protocol lies in the reliable transfer it offers. To avoid network collapse, TCP reacts to congestion by reducing its rate. This reduction depends on the way the loss, which is used as indication of congestion, is detected. If the loss is detected by duplicate acknowledgments (typically 3: RFC-2581), the congestion window is halved. Otherwise, the loss is detected by timeout and the sender reduces the size of its congestion window to one packet. Depending on the way the loss is detected, TCP enters a slow-start phase (in the case of timeout) or congestion avoidance phase (in the case of triple duplicates). The way the sender increases its congestion window is also phase dependent. More details can be found in RFC-1122, RFC-2581 and in [3].

The large usage of TCP makes it benefit from a special attention. Many studies have been done to understand its behaviour and the parameters it depends on. Several analytical models (e.g. [11], [14], [10]) have been developed for the throughput of long-term TCP connections and have helped understand the impact of certain parameters. However, these models have been obtained under different assumptions and all assume that the phase of fast recovery is negligible and that the source resumes the linear increase of its congestion window directly

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 488–500, 2006.

after the reduction (as pointed by Altman et al. in [1]). More sophisticated models exist that try to alleviate some of these hypotheses. For example, [9] and [1] take into account the effect of the window size on the round-trip time and also the correlation between losses. The model proposed in [15] is more accurate and furthermore it takes into account the slow start, which makes it usable also for short sessions.

Another consequence of the success of TCP is that any new protocol deployed on the internet should be TCP-friendly [6] in order not to disturb 90% of the traffic. To reach this TCP-friendliness, some multicast and real time protocols (e.g. [16], [18], [7], [5]) have used the analytical models of the TCP throughput, to adapt their rate so as to obtain similar throughput as TCP in the same network conditions. The most popular models for these applications are the SQRT [11] and the PFTK [14] formula.

According to these two models, the TCP throughput is inversely proportional to the round-trip time (at least in the case of low loss rate). This statement has an important consequence: if two TCP sessions share the same bottleneck then the ratio between their throughput (in terms of packets) is equal to the inverse of the ratio between their round-trip times. A simple experiment can however show that this statement is not always true. To this end, we ran different scenarios (100) with a simple topology consisting of one bottleneck over which a certain number of TCP New-Reno sessions compete. They all have the same packet size and different round-trip times. In each simulation, the bandwidth of the bottleneck and the number of concurrent sessions are chosen randomly and the loss rate is low (under 2%). Over all the scenarios, we record for each pair (i, j) of TCP connections the ratio between their throughput $(\frac{B_i}{B_j})$ and the inverse ratio between the average of their round-trip times $(\frac{RTT_j}{RTT_i})$. All the pairs $(\frac{RTT_j}{RTT_i}, \frac{B_i}{B_j})$ are represented by a point in the scatter plot of Figure 1. According to the models, each point should be on the line $y = x$ since the two ratios should be equal. Figure 1 shows that even in the case of simple topologies, the scatter plot is not fitting the model. In some cases, the ratio between the throughput equals seven times the inverse ratio between the average round-trip times. This also means that TCP is not always fair towards other TCP sessions. In [2], the authors have observed this unfairness in short term sessions.

In this paper, we propose to study the accuracy of the SQRT and PFTK models that are recalled in Section 2. The approach we propose for the validation is based on gathering an important number of TCP sessions obtained in different randomly generated topologies and scenarios, and comparing the throughput really obtained with the predicted one. The way we proceed and the results of the validation are given in Section 3. We show then, in section 4, that the main reason of the inaccuracy of the two models is the aggregation of the losses, which is also present in other TCP models [1, 9, 12, 13]. To the best of our knowledge, none of the previous work has made distinction between the two loss rates. In Section 5, we show that taking into account the timeout loss rate can greatly improve the accuracy of TCP throughput models obtained by supervised learning techniques. Finally, we conclude in Section 6.

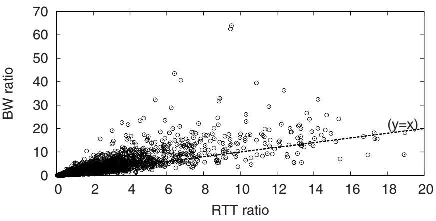


Fig. 1. The ratio of 2 session rates versus the inverse ratio of their RTTs

2 Analytical Models of TCP Throughput

In this section, we give a brief reminder of the PFTK and SQRT models which are often used by other protocols to offer TCP-Friendliness. Both have been developed for long-term TCP connections (i.e., for flows with a large amount of data to send, such as file transfers).

In 1997, the only formula modelling the throughput of TCP was the one developed by Mathis et al. in [11] which is:

$$B_{tcp} = \frac{C \cdot MSS}{RTT\sqrt{p}} \tag{1}$$

where $C \approx 1.22$, MSS is the maximum segment size, RTT the average round-trip time, and p the loss rate over the session. This model is often called the SQRT model. Only the congestion avoidance is taken into account in this model and all the losses are assumed to be detected by triple duplicates. This assumption implies that the loss rate is low and that there is no timeout expiration.

In 1998, Padhye et al. [14] have developed a more complex formula, taking into account losses detected by timeouts which are frequent in high loss rate environments. The details and the hypothesis made for this model can be found in [14]. The formula, called PFTK, is summarised as follows:

$$\begin{aligned} &B_{tcp} \approx min(\tfrac{rwnd}{RTT}, \tfrac{MSS}{RTT\sqrt{\frac{2bp}{3}}+f(p)}) \\ &\text{with } f(p) = T_0 min(1, 3\sqrt{\tfrac{3bp}{8}})p(1+32p^2) \end{aligned} \tag{2}$$

where MSS, p and RTT are as described in the SQRT formula. T_0 is the initial value of the timeout, $rwnd$ the receiver window and b the number of packets acknowledged at once. This formula is said in [14] to be developed for Reno, but it is based on the hypothesis that all the packets sent after the loss and belonging to the same window are lost. This hypothesis means that the sender can only decrease its congestion window once by round-trip time, which is true only in the case of NewReno (see [1]).

The two formulas do not take the effect of slow-start phases into account, which makes them unusable for the prediction of the throughput of short TCP sessions. They also neglect the fast recovery phase as said previously. Other assumptions have also been made for the modelling but we do not develop them in the paper. The reader can refer to the original papers and to [1] or [17] for more detailed discussions of these models.

3 Models Validation

In this section, we propose to validate the two models (SQRT and PFTK) by using a generic approach based on random simulations. More precisely, the quality of the models is measured by their ability at predicting the throughput of TCP in various topologies and scenarios. The way we generated these topologies and scenarios is described in Section 3.1. In Section 3.2, we give the criteria we use to measure the quality of model predictions. The results of the validation of the two formulas are discussed in Section 3.3.

3.1 Topologies and Scenarios Used

To validate the SQRT and PFTK formulas, we use 7600 TCP New-Reno[1] sessions chosen randomly over thousands of topologies with different scenarios. Since the two formulas are developed for long-term throughput, we choose TCP sessions that last at least 400 seconds and for which the time needed to send all packets in a window is smaller than the RTT. The receiver window is chosen very large so as not to be the bottleneck. To create a topology and a scenario we proceed as follows: a network topology is generated randomly and then the network is simulated during a fixed amount of time, again by generating the traffic randomly. At the end of the simulation, we collect for all TCP New-Reno sessions that last at least 400 seconds, the loss ratio p computed over the whole session, the value of Maximum Segment Size *(MSS)*, the value of the timeout T_0, the average round-trip time RTT, the number of packets acknowledged at once b, and the TCP throughput obtained. This procedure is repeated until we have a sufficient number of sessions in the database.

To generate a random topology, we first select a random number of nodes (between 10 and 600) and then choose randomly the connections between these nodes. The bandwidth, the propagation delay, and the buffer size of the links were chosen randomly. The bandwidth is chosen between 56Kb/s and 100Mb/s while the propagation delay varies between 0.1ms and 500ms.

Concerning the traffic, the flows were chosen randomly among TCP and other types of traffic based on UDP and proposed by `ns-2`. The senders, the receivers, and the duration of each traffic were set randomly.

We are aware that this large set of random topologies and traffic conditions may also include many non realistic ones, but since the various analytical models of TCP are topology and traffic unaware, they are not supposed to give good results only in realistic scenarios. Moreover, as characterizing realistic scenarios

[1] We have used TCP NewReno, and not Reno, because, as stated earlier, both formulas are based on the assumption that the congestion window can only decrease once per RTT.

is beyond the state of the art, trying to restrict ourselves to a large set of so-called realistic scenarios may create the risk of being too restrictive and thus introduce a bias.

3.2 Evaluation Criteria

The quality of a model, or of an estimator, of which the goal is to predict a numerical output from some inputs, depends on how well it fits the data to predict. The closer the predicted value to the observed one, the more accurate is the model. Its accuracy, or its adjustment, can be measured by different statistics. The mean square error is often used as well as the coefficient of determination. The mean square error is equal to:

$$MSE = \frac{1}{N}\sum_{t\in\tau}(\hat{X}_t - X_t)^2, \tag{3}$$

where τ is the set of data to predict of size N, X_t is the value to estimate, and $\hat{X}_t$ the value estimated by the model. The coefficient of determination is defined as:

$$R^2 = \frac{\sum_{t\in\tau}(\hat{X}_t - \overline{X})^2}{\sum_{t\in\tau}(X_t - \overline{X})^2} = 1 - \frac{\sum_{t\in\tau}(\hat{X}_t - X_t)^2}{\sum_{t\in\tau}(X_t - \overline{X})^2} \tag{4}$$

where $\overline{X}$ is the average of X_t over τ. Note that $(\frac{\sum_{t\in\tau}(\hat{X}_t - X_t)^2}{\sum_{t\in\tau}(X_t - \overline{X})^2})$ is the ratio between the mean square error and the variance.

The closer the coefficient of determination to 0, the more the scatterplot is spread around the regression line, which means the less accurate the model is. And inversely, the lower the amount of spread points around the regression line, the more the coefficient of determination is close to 1 (a perfect fit).

Since PFTK and SQRT formulas are used to offer TCP-Friendliness, the ratio between the predicted value and the value to predict is another important measure of the model accuracy. This ratio should be close to one so that a protocol using one of the two formulas to determine its rate can provide TCP-Friendliness. This information is not carried in the MSE or in the coefficient of determination that are only sensible to absolute distances between the predicted and real values. The ratio can be very high whereas the distance between them can be very small (in comparison with the average distance between the predicted values and the values to predict in the set τ).

We thus need another criterion that takes this ratio $(\frac{\hat{X}}{X})$ into account. The maximum (resp. the minimum) of this ratio gives an idea of how much the prediction can overestimate (resp. underestimate) the true value. However, the average and standard deviation of the ratio should be analysed with caution. Indeed, underestimation affects the average and the standard deviations less than overestimation while both have the same importance for our application. Therefore, in addition to the ratio, we propose to use also the *"absolute ratio"* defined by $max(\frac{\hat{X}}{X}, \frac{X}{\hat{X}})$, and which is sensitive in the same manner to both underestimation and overestimation.

Subsequently, we will use the coefficient of determination as well as the two ratios to evaluate TCP throughput estimators.

3.3 SQRT and PFTK Accuracy

For each session of the validation set (consisting of 7600 TCP sessions), we compute, based on the collected parameters, the throughput predicted by SQRT and PFTK. We then compute for the set of predicted data, the coefficient of determination, and several statistics (average, minimum, maximum and standard deviation) concerning the two ratios ($^{\hat{X}_t}/_{X_t}$ and $\max(^{\hat{X}_t}/_{X_t}, ^{X_t}/_{\hat{X}_t})$).

We plot in Figure 2 the predicted throughput as a function of the real throughput (the one we try to predict) for the two models. Ideally, the scatter plot should fit the regression line ($y = x$), i.e. the predicted value should be equal to the value to predict. Figure 2 shows that both models are far from fitting the regression line. A great amount of points are in fact spread around the latter. Figures 3 and 4 show the ratio and the absolute ratio with respect to the real throughput. In the ideal case, both graphs should be reduced to the straight line $y = 1$. This is again far from being the case. Whatever the performance criterion, both models are inaccurate. Moreover, the range of the ratio in the case of SQRT

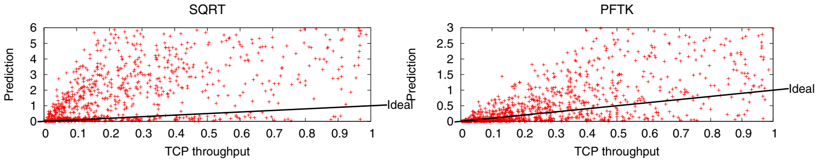


Fig. 2. The predicted throughput versus the real throughput

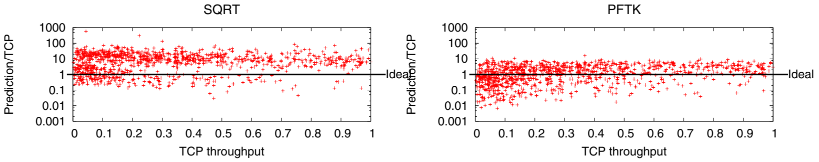


Fig. 3. The ratio between the predicted throughput and real throughput versus the real throughput

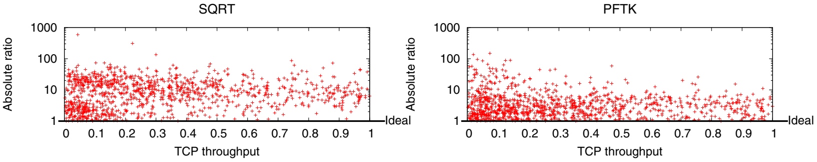


Fig. 4. The absolute ratio between the predicted throughput and real throughput versus the real throughput

Table 1. The coefficient of determination (R^2). the mean square error (MSE) and statistics of the ratio (R) and the absolute ratio (AR) of SQRT and PFTK models.

	R^2	MSE 10^{-3}	R				AR			
			avg	stdev	min	max	avg	stdev	min	max
Mathis	0.658	4.078	5.29	10.29	0.013	583.73	5.69	10.59	1	583.73
PFTK	0.814	2.211	2.2	1.19	0.006	16.11	3.15	5.81	1	152.59

is $[0.013, 583.73]$ while the range of PFTK is $[0.006, 16.11]$. That means that a protocol that uses the SQRT model to provide the fairness can get 583.73 times more than what a concurrent TCP would get, and 152.59 (= 1/0.006) times less when using PFTK. In both cases, the protocol would not be TCP-Friendly.

Table 1 summarises the above figures numerically. The table shows the coefficient of determination and some statistics concerning the normal ratio and the absolute ratio of the two models. The SQRT model is less accurate than the PFTK one, which has already been shown in [14]. In other words, a protocol that will use the SQRT model will be less TCP-Friendly than one using the PFTK model. However, even if PFTK shows a better behaviour than SQRT, it is still not TCP-Friendly. The absolute ratio should be lower than 1.78 (as suggested in [7]) to provide the fairness towards TCP. Its average is 5.69 for SQRT against 3.15 for PFTK, which are both above 1.78. More precisely, over our validation set (7600 TCP sessions), 70% of PFTK predictions are not TCP-Friendly.

So, in conclusion, neither the SQRT model nor the PFTK model is accurate. This is due *in part*[2] to the fact that phases like slow-start and fast-recovery are not taken into account and that many hypotheses have been made to make the derivation of an analytical formula feasible[1, 17].

4 Analysis of Bad Predictions

In the previous section, we have shown that SQRT and PFTK are not always accurate and even more, the throughput they predict is often not TCP-Friendly. In this section, we propose to investigate the reasons for the bad predictions of the models. This study aims at characterising the conditions leading the formulas to under or overestimate the throughput. To this end, we propose to use an original approach based on the analysis of the validation test by the decision tree method (which provides *"interpretable"* models). With this method, we will build a classification model to discriminate the good and the bad predictions in function of different parameters gathered from the network. The analysis of the tree so obtained will then provide a characterisation of the conditions under which the models give bad predictions. Before going to the analysis in Section 4.2, we first explain how we classify the predictions into good and bad predictions.

[2] We will see latter that other causes exists.

4.1 Classification of Predictions

The model prediction will be considered as good if it preserves TCP-Friendliness. In other words, a prediction is considered as "good" if the ratio between itself and the true throughput it approximates belongs to $[1/K, K]$, where $K \geq 1$ is a factor that defines the bounds of fairness towards TCP. Out of this interval, the ratio can belong to $(0, 1/K)$ and in this case the prediction underestimates the throughput to predict, or the ratio belongs to (K, ∞) which is a case of overestimation. These three areas are represented graphically in Figure 5.

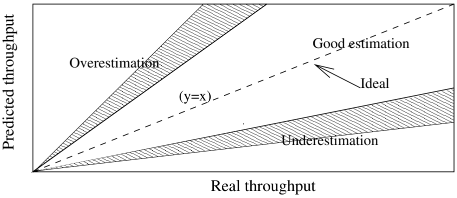


Fig. 5. The three areas defining the quality of the predictions of a model

To define bad predictions, we choose another parameter $K' > K$ and we consider as bad predictions the predictions that are such that the ratio belongs to $(0, 1/K') \cup (K', +\infty)$. This definition of bad predictions thus leaves a region of fuzziness between good and bad predictions which is represented in grey in Figure 5. Subsequently, by abuse of language, the word overestimation (resp. underestimation) will be used to denote the overestimation (resp. underestimation) area of the graph minus the gray area. Thus, the words underestimation and overestimation will be synonyms of bad prediction.

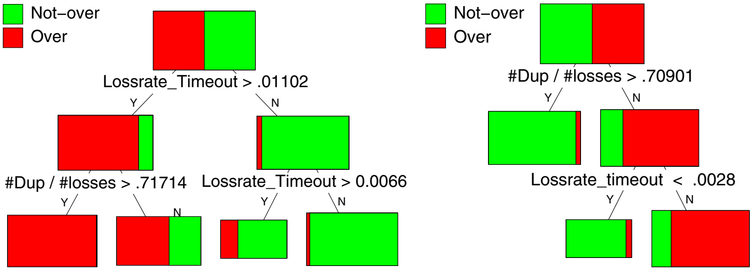
For our study, we use the commonly accepted value of $K = 1.78$ to define TCP-Friendliness and we consider that if the absolute ratio is higher than $K' = 3$, then the prediction is bad. The value of this threshold is purely subjective. However, a study had been done with a threshold equal to 10 and had led to similar results. The choice of the value three is thus not restrictive.

4.2 Decision Trees Analysis

As said in the previous section, there are two kinds of bad predictions: underestimation and overestimation. We have then 4 cases to analyse: PFTK and SQRT in both the underestimation and then overestimation case. Table 2 shows the

Table 2. Distribution of prediction types for the two models

Interval	PFTK	SQRT
$[0.001, 1/3)$ (under)	6.14%	3.43%
$[1/3, 3]$	33.42%	25.66%
$(3, 1000]$ (over)	60.44%	70.91%



(a) SQRT overestimation (b) PFTK overestimation

Fig. 6. The top of the decision trees classifying bad predictions. (lossrate_timeout is defined as the number of losses detected by expiration timeout (#timeout) divided by the number of packets transmitted).

distribution of the predictions of PFTK and SQRT into the different classes. The cases of underestimation are too rare for both PFTK and SQRT to obtain statistically meaningful conclusions from their analysis. So, we will drop these cases. It thus remains two situations: SQRT and PFTK in overestimation.

To analyse the reason for overestimation in both cases, we first classify each prediction into one of two classes: *"Over"* to denote the predictions that overestimate the TCP throughput and *"Not-Over"* to denote the other predictions. Then a decision tree is built to explain the classification using as inputs the PFTK parameters, the proportion of losses detected by triple duplicates, and the proportion of losses detected by timeout expirations. The top of each tree is represented in Figure 6 and is discussed below.

The tree of Figure 6(a) shows that if the proportion of losses due to timeout expiration exceeds a certain threshold then SQRT overestimates the throughput. Indeed, at each timeout loss no data is transfered, and this sender inactivity is not taken into account by the model. The model still considers that data are sent and the predicted throughput obtained is higher than TCP's. This result is already known in the networking community.

The tree of Figure 6(b) is related to the overestimation of PFTK. It points out that if the proportion of losses due to triple duplicates is under a certain threshold, then the estimation exceeds the real throughput. When the number of losses detected by triple duplicates decreases, the number of losses detected by timeout increases. The loss rate p used in $E[A]$ (eq. (16) of [14]) becomes then higher than the value that should be used since it should include only losses due to triple duplicates. Thus, $E[A]$ is lower than what it should be and the predicted throughput, B (inversely proportional to $E[A]$), is then higher than the real one. In addition, when the number of timeouts increases, the number of slow-start phases not taken into account, and over which the throughput predicted is higher than the throughput to estimate, increases. The timeout loss rate affects also the prediction of PFTK.

In conclusion, a discrimination between the way the losses are detected seems to be required for a good prediction. To the best of our knowledge, no models of

TCP, even recent models such as [1] or [15], make this distinction. In the next section, we highlight the importance of incorporating the timeout loss rate in the context of models inferred by machine learning techniques.

5 Supervised Learning of TCP Throughput Models

In this section, we propose to use supervised learning methods to infer models to predict the TCP throughput. We choose these methods because, unlike analytical models, they do not make any assumption about the network and protocol. These methods automatically build a model of an input/output relationship solely from a database of observations of input/output pairs. As incorporating new inputs in these models is straightforward, we begin by inferring models of the TCP throughput using as inputs the same parameters as in PFTK formula[3] and in a second step, we introduce the timeout loss rate in addition to these parameters. The comparison of these two classes of models will highlight the importance of distinguishing the two loss types.

The database used to infer models contains 18000 sessions generated by using the procedure described in Section 3.1. These models are then evaluated on the (independent) validation set of 7600 sessions used to evaluate the analytical models. In this paper, we present the results obtained by two machine learning algorithms: Multiple Additive Regression Trees (MART) and Multilayer Perceptrons (MLP). The interested reader can refer to [4] for more details about database generation and machine learning methods.

Table 3 compares the analytical models to the two machine learnt models, with and without the timeout loss rate (TLR). As in previous sections, we compute for each model the coefficient of determination, the mean square error as well as statistics related to the ratio and the absolute ratio.

Table 3. The coefficient of determination (R^2). the mean square error (MSE) and statistics of the ratio (R) and the absolute ratio (AR) of MART and MLP with and without the timeout loss rate (TLR).

		SQRT	PFTK	MART		MLP	
				Without TLR	with TLR	without TLR	with TLR
$MSE 10^{-3}$		4.078	2.211	1.246	0.482	1.048	0.322
R^2		0.658	0.814	0.895	0.960	0.912	0.973
R	avg	5.29	2.2	1.23	1.11	1.62	1.12
	min	0.013	0.006	0.07	0.16	0.05	0.2
	max	583.73	16.11	25.91	6.49	18.9	5.2
	stdev	10.29	1.19	0.7	0.49	2.2	0.54
AR	avg	5.69	3.15	1.46	1.22	1.78	1.24
	min	1	1	1	1	1	1
	max	583.73	152.59	25.91	6.49	20	5.2
	stdev	10.59	5.81	0.83	0.55	2.14	0.58

[3] These parameters are also used in [15].

We first notice that, on our validation set, supervised learning models are much more accurate than analytical models, whatever the set of parameters used. They reduce the mean square error by more than a factor two and they offer much better fairness towards TCP than the analytical model. Furthermore, the introduction of the timeout loss rate significantly improves their accuracy. The mean square error has been reduced by about a factor three for both methods and the coefficient of determination is much closer to one. In terms of fairness, the average absolute ratio is reduced from 1.46 to 1.22 in the case of MART and from 1.78 to 1.24 in the case of MLP. The reduction of the average is not as important as it is in the case of the maximum absolute ratio. Indeed, the maximum goes from 25.91 and 20 for MART and MLP respectively, down to 6.49 and 5.2. The positive impact of the introduction of the timeout loss rate is clear from this experiment and makes us believe that analytical models will also benefit from the introduction of this parameter.

Our machine-learnt models of TCP have been derived from a large set of random topologies and traffic conditions. The randomness of the learning set may be reduced by focusing more on so-called realistic topologies and traffic conditions, provided that good criteria are found to characterize them. Note however that doing so can only improve our learnt models, because the realistic cases, being a subset, will obviously drive the learning algorithms to more specialized models. A random learning set is thus actually a worst case for our approach. As our models are already quite better than existing analytical models in such a worst case, our approach and results are therefore already very encouraging.

6 Conclusions

In this paper we have studied the accuracy of SQRT and PFTK models. To this end, we have built a database with a high number of TCP sessions gathered in random scenarios and have compared the results predicted by the models with the observed throughputs. Neither SQRT nor PFTK is accurate and we have pointed out the reason of that. PFTK, which is the reference among the analytical models of TCP throughput uses the global loss rate p that accounts indifferently for losses detected by triple duplicates and losses detected by timeout expirations. This non discrimination affects the result: the throughput is overestimated. The application of machine learning algorithms allows us to highlight the importance of the distinction of the two types of losses, which can indeed greatly improve the quality of the models. Our analysis suggests also that future research aiming at the analytical modelling of the throughput of TCP should certainly take into account the timeout loss rate.

We have also proposed an alternative to analytical modelling based on supervised learning, which offers better results than the two tested models even without discriminating the losses. In the future, it would be very interesting to compare these models with more recent TCP models such as those of Altman et al. [1] and of Sikdar et al. [15]. However, the first one uses an infinite sum of terms which is difficult to compute in practice while the second one has been developed for Reno and not NewReno. On a broader point of view, we would like to further exploit machine learning techniques in networking. These methods do not make any hypothesis and can take implicitly into account all TCP phases and long as well as short sessions, provided that they are represented in the database. The approach can also be easily extended to any version of TCP or any other protocol. Finally, the application of supervised learning techniques is automatic and needs much less time and effort than an analytical modelling, which may be of great importance in the rapidly evolving domain of networking.

Acknowledgements. This work has been partially supported by the Belgian Science Policy in the framework of the IAP programme (MOTION P5/11 project) and by the E-NEXT European Network of Excellence (NoE). P.G. is a Scientific Research Worker at FNRS, Belgium.

References

1. E. Altman, K. Avrachenkov, and C. Barakat. A Stochastic Model of TCP/IP with Stationary Random Losses. *IEEE/ACM Transactions on Networking*, 13(2):356–369, April 2005.
2. F. Baccelli and D. Hong. AIMD, Fairness and fractal scaling of TCP Traffic,. In *IEEE INFOCOM 2002*, volume 21, pages 229 – 238, June 2002.
3. C. Barakat. TCP/IP modeling and validation. *IEEE Network*, 15(3):38–47, 2001.
4. I. El Khayat, P. Geurts, and G. Leduc. Analysis and improvement of analytical models of TCP throughput by machine learning techniques. Technical report, University of Liège, 2005.
5. I. El Khayat, P. Geurts, and G. Leduc. Improving TCP in wireless networks with an adaptive machine-learnt classifier of packet loss causes. In *Proc. of the International Conference on Networking*, pages 549–560. Springer-Verlag, 2005.
6. S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Trans. Netw.*, 7(4):458–472, 1999.
7. S. Floyd, M. Handley, J. Padhye, and Jorg Widmer. Equation-based congestion control for unicast applications. In *SIGCOMM 2000*, pages 43–56, Stockholm, Sweden, August 2000.
8. C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S.C. Diot. Packet-level traffic measurements from the sprint ip backbone. *Network, IEEE*, 17(6):6– 16, Nov.-Dec. 2003.
9. M. Garetto, R.L. Cigno, M. Meo, and M. A. Marsan. Closed queueing network models of interacting long-lived TCP flows. *IEEE/ACM Trans. Netw.*, 12(2):300–311, 2004.
10. A. Kumar. Comparative performance analysis of versions of TCP in a local network with a lossy link. *IEEE/ACM Trans. Netw.*, 6(4):485–498, 1998.
11. M. Mathis, J. Semke, Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM Computer Communication Review*, 27(3):67–82, July 1997.
12. A. Misra and T. J. Ott. The window distribution of idealized TCP congestion avoidance with variable packet loss. In *INFOCOM (3)*, pages 1564–1572, 1999.
13. Vishal Misra, Wei-Bo Gong, and Donald F. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *SIGCOMM*, pages 151–160, 2000.

14. J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *Proceedings of the ACM SIGCOMM '98*, pages 303–314, 1998.
15. B. Sikdar, S. Kalyanaraman, and K. S. Vastola. Analytic models for the latency and steady-state throughput of tcp tahoe, reno, and sack. *IEEE/ACM Trans. Netw.*, 11(6):959–971, 2003.
16. D Sisalem and A Wolisz. MLDA: A TCP-friendly congestion control framework for heterogenous multicast environments. In *Eighth International Workshop on Quality of Service (IWQoS 2000)*, Pittsburgh, June 2000.
17. J. Widmer, R. Denda, and M. Mauve. A Survey on TCP-Friendly Congestion Control. *IEEE Network*, 15(3):28–37, 2001.
18. J. Widmer and M. Handley. Extending equation-based congestion control to multicast applications. In *Proceedings of SIGCOMM'01*, pages 275–285. ACM Press, 2001.

High Speed Packet Logging on a Budget

Chad D. Mano, Jeff Smith, Bill Bordogna, and Aaron Striegel*

Department of Computer Science and Engineering,
University of Notre Dame,
Notre Dame, IN 46556, USA
{cmano, jsmith30, wbordogn, striegel}@nd.edu

Abstract. Creating high quality network trace files is a difficult task to accomplish on a limited budget. High network speeds may overburden an individual system running packet logging software such as tcpdump, resulting in trace files with missing information and making analysis difficult or incomplete. High end specialized systems may perform the job well, but may be out of reach due to financial constraints. To the end we developed the *Cheap Logger* (CLog) system which utilizes inexpensive COTS hardware to create a high quality, complete network trace files. A scalable distributed storage system enables the CLog system to expand and continue to create high quality, complete network data trace files even at extremely high data rates.

1 Introduction

An important aspect of some areas of computer network research is the ability to perform analysis in a large-scale network environment. However, most network administrators would be reluctant to allow an experimental device to be incorporated into a live enterprise network, particularly in critical areas such as the gateway to the Internet. This leaves laboratory simulation as the only remaining option to perform large-scale network analysis.

Software packages such as *ns-2* [10] are useful for simulations where performance is the key issue being addressed, but falls short in the ability to create a highly accurate representation of actual data flow within a large scale network. An accurate representation of traffic is essential where packet payload analysis is needed such as in virus and worm detection research [8, 11].

A solution to the need for an accurate representation of network traffic is to capture and store live network data. On a small scale this can be easily accomplished with a standard desktop computer and an application such as tcpdump [6]. However, as network capacity increases it becomes difficult to keep up with line speeds resulting in an incomplete network trace. Powerful systems designed for high speed packet logging are available, but may break the budget of a research group [1, 5].

* This research was supported in part by the National Science Foundation through the grant CNS03-47392.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 501–512, 2006.

This financial hurdle led to our development of a high speed network packet logger which can be built for a fraction of the cost of a commercial system. The *Cheap Logger* (CLog) system is built from inexpensive hardware and can easily be scaled to meet increased demands for logging speed. This paper describes the CLog system architecture, associated utilities, and presents a performance analysis of the system.

The remainder of the paper is organized as follows. Section 2 briefly presents the motivation and background for the project. Section 3 details the architecture of the system including communication protocols developed for management of the system. Section 4 analyzes the performance capabilities of the system. Finally, Section 5 summarizes the work.

2 Motivation and Background

For various endeavors in our research group, a trace of live network traffic is needed to measure the performance and scalability of systems which have been developed. Live data is accessed via a tap of the University of Notre Dame Internet gateway. The network tap comes from a fiber gigabit link at the edge of the Notre Dame network which feeds an OC-12 line to the University's ISP via multiple 100Mb/s connections. The tap point averages just over 130Mb/s utilization.

In the past, an HP zx2000 Itanium2 workstation was used as the packet logging system. Even this relatively fast desktop system would drop a significant number of packets leaving an incomplete trace file of network traffic. The workstation was assumed to be powerful enough to keep pace with the speed of the Internet tap and through simple experimentation and analysis this proved to be the case. The bottleneck, it was discovered, proved to be the hard disk (15k rpm SCSI drive) which could not keep up with amount and speed of the traffic coming from the tap.

To ease the disk write speed problem, we developed a solution that distributes data writing tasks over multiple systems, thus relieving the bottleneck created by a single system logger. Two important requirements for the system is that it should be inexpensive and scalable. The components of the system are typical *commercial-off-the-shelf* (COTS) hardware, making the creation of the system affordable. The client/server architecture makes the system scalable as additional inexpensive clients may be dynamically added to the system as network capacity increases.

3 Architecture

The physical architecture of the CLog system is a standard client/server model. The server acts as the gateway for the network tap and forwards data to be logged to each individual client system as illustrated in Figure 1. A Sun dual Opteron 244 workstation with 1GB or RAM was utilized as the server for CLog. It was desireable for the clients to be physically small as there would be multiple client machines and the entire CLog system needed to fit on a cart that could

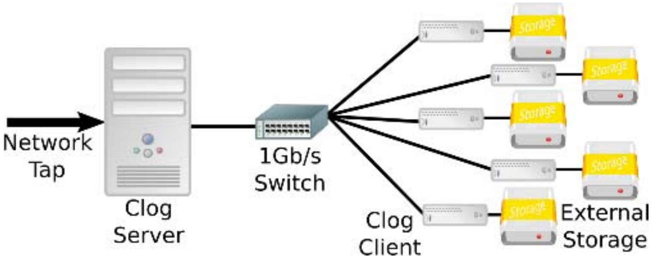


Fig. 1. Illustration of the Cheap Logger system

be moved from place to place. Therefore, Apple 1.25 GHz Mac mini systems running Darwin Kernel Version 1.9.0 were utilized as the client machines for the system. An external LaCie Firewire 7200 RPM 500GB hard drive was attached to each Mac mini for data storage.

The remainder of this section details the individual components of the CLog system and the communication protocols designed to maintain data consistency and incorporate management capabilities.

3.1 Server

The server provides the central control of the system and is the gateway for the network tap data. The server must have at least two Ethernet ports and a third if remote access is required to operate the system. Remote management is simply performed by establishing an SSH session from a different system. We utilize the on-board 1Gb/s Ethernet port for remote management and installed an Intel Pro/1000 MT dual port 1Gb/s NIC for the required ports.

The data flow of the network tap traffic is one-way as the CLog system does not inject data back into the network. Therefore, the port which is connected to the network tap is designed as the *inbound port* (i-port) and the remaining port is designated the *outbound port* (o-port). The o-port is connected to a 1Gb/s switch which, in turn, is connected to each client system.

The processing of the network tap data is minimal as the server must keep up with the speed of the incoming packets. The data flow of within the server is illustrated in Figure 2. When a packet is captured on the i-port the server immediately writes the packet to tail point of a cyclical buffer. A separate thread removes packets from the head point of the buffer and overwrites the destination MAC address in the Data Link header of the packet. The packet is then transmitted on the o-port and is forwarded on to one of the client systems.

It is possible to implement a load balancing scheme to enhance the overall effectiveness of the distributed writing system. However, in an effort to minimize the processing requirements of the server a simple round-robin system is utilized to determine the destination client of each packet. The round-robin method is essentially a *least recently used* queue of all client systems.

In cases where analysis is only concerned with capturing complete packet traces this process is sufficient. However, in most cases exact ordering of packets

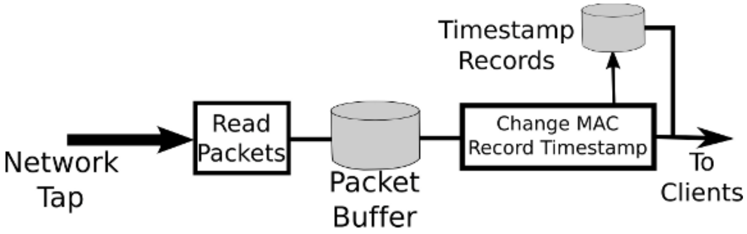


Fig. 2. Diagram of the data flow within the Cheap Logger server

and accurate timing records are essential. This is accomplished by the server taking additional processing steps. This process will be discussed following the introduction of the client system.

3.2 Client

The clients perform the "physical labor" of the system by writing the network data to disk. A custom packet logger application, similar to tcpdump, was developed using the Libpcap [4] C library. The packet logging process is discussed here with the server communication for management purposes to be presented later in this section.

After a client is connected to a server it is able to begin logging packets. In order to do this efficiently a packet buffer and a multi-threaded disk writing process is utilized as illustrated in Figure 3. Each incoming packet is stored in a single packet buffer. A parameter setting determines the maximum size of the buffer based on the number of packet contained in the buffer. When the buffer reaches the packet number limit a new thread is created which takes the data in the packet buffer and writes it to disk.

The main thread is able to continue to capture new packets while the disk writing takes place. The new packet writing thread is placed in a queue of all packet writing threads. A new thread is created each time the packet buffer is filled to prevent packet loss which may occur if a single thread was expected to perform all writing responsibilities. In such a case the thread may be busy writing and unable to collect data from the packet buffer at the instant it is needed.

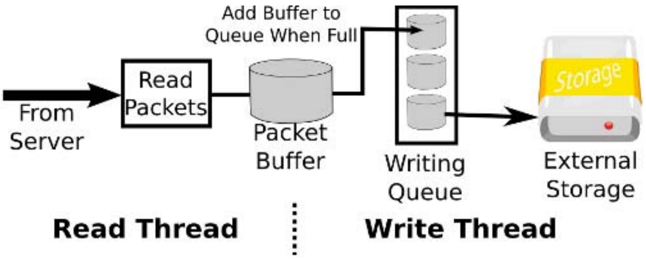


Fig. 3. Illustration of the Cheap Logger client system

The log files are stored in tcpdump format. This allows the final trace files to be analyzed using existing applications such as such as tcpdump or ethereal [2]. As part of this logging format, a timestamp header is stored with each packet. The timestamp is important for any timing related analysis or to simulate the data flow at a later time using an application such as *tcpreplay* [7]. It is critical, therefore, to maintain precise timing data for the entire system. This property is maintained by the timestamp synchronization process.

Timestamp Synchronization. Timestamps are created when a new packet arrives on a packet logging system using the libpcap library. In the Clog system this results in two timestamps being associated with each packet. First, when a packet is received by the server a timestamp is created. This timestamp is the most precise in relation to other packet timestamps as all timestamps at this point are based on a single clock and are recorded prior to processing within the CLog system. The second timestamp is recorded on the client systems, each using the local clock to determine the timestamp value.

This second timestamp is not useful as clocks most certainly vary slightly, even if attempts are made to synchronize the clocks with a common time server [9]. In addition, the timestamps generated on the clients are created after the packets have spent a non-trivial amount of time due to processing and transmission in the server system. The time synchronization solution requires the original timestamps to ultimately replace the timestamps recorded on the client systems.

Libpcap headers (including timestamps) are prepended to the packets and are not actually part of the raw packet. Therefore, timestamps cannot simply be added to each existing packet header prior to forwarding the packet to the client. In addition, appending timestamps as a footer to the payload of each packet is not possible due to MTU restrictions. Even without the MTU restriction, the overhead of updating existing packet header information (size information and checksums) may prove to create a new bottleneck for the system.

Our solution was to log timestamps as generated on the server and periodically forward the log to the associated client. Figure 4 illustrates this process. When a

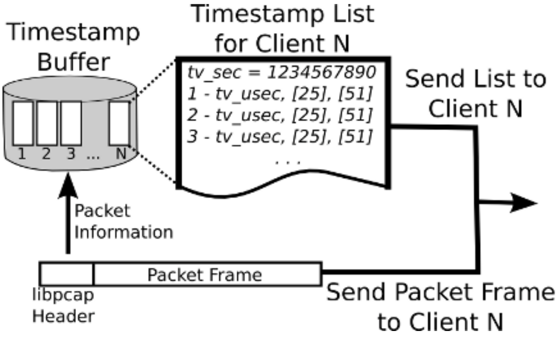


Fig. 4. Illustration of the timestamp system

packet is received by the server, a timestamp is immediately generated. Once the destination client has been determined for the packet the timestamp is recorded in a buffer. An individual timestamp consists of two parts, the number of seconds (*tv_sec*) since the Epoch (00:00:00 UTC, January 1, 1970) and the number of additional microseconds (*tv_usec*). When the first timestamp is recorded in the log, four items are recorded. The value *tv_sec* is recorded as a global value for the log meaning this value is stored only once for all packets which will be part of the log. Next the value *tv_usec* is recorded along with two additional bytes from the header of the packet. These last three items are logged for each packet added to the log.

The two bytes are used as a simplistic method to match the timestamp with the appropriate packet stored on the client. The bytes are designated as an offset from the beginning of the packet frame and through empirical analysis values of 25 and 51 were found to be effective. An analysis of the network trace shows that in 99.92% of the time the combination of these two bytes are unique from the neighboring ten packet frames. The packets and timestamps are stored sequentially, therefore, any ambiguity between packet frames in which these two bytes match is easily removed by comparing neighboring values.

A single timestamp log is forwarded on to a client if either the log becomes full or if a timeout period expires. The timeout period is essential as the log stores a global *tv_sec* value for all packets in the log. The timeout period is set to be less than one second which restricts a log from containing anything else besides packets with an identical *tv_sec* value or a value one second greater than the first packet stored in the log. This allows a post-processing step to determine, without ambiguity, the exact timestamp of the packet.

The Libnet [3] C library is used to create the timestamp log packet to send to the client. The *Ethertype* field of the Ethernet header is modified to a custom value enabling the post-processing step to identify timestamp log packets. The post-processing procedure can be implemented in one of two ways. First, the timestamps can be corrected on the trace files from a single client. A packet sorter can then be applied later to merge the files from all clients. The other option is to process trace files from all clients simultaneously which results in new trace files containing the merged data from all files.

3.3 Communication

Communication between the server and clients is required for two purposes: for the discovery of clients in the system, and for statistical updates throughout the capture. This communication protocol is efficiently implemented using the libnet library and handles all communication via Ethernet addressing and therefore does not require IP layer processing. The basic communication structure is illustrated in Figure 5.

A client, when ready to begin logging, broadcasts an announce (ANN) message and continues to do so until a response is received. The server listens on the o-port for ANN messages and responds with an acknowledgment (ANN_ACK). The server then adds the MAC address of the client to the LRU queue for

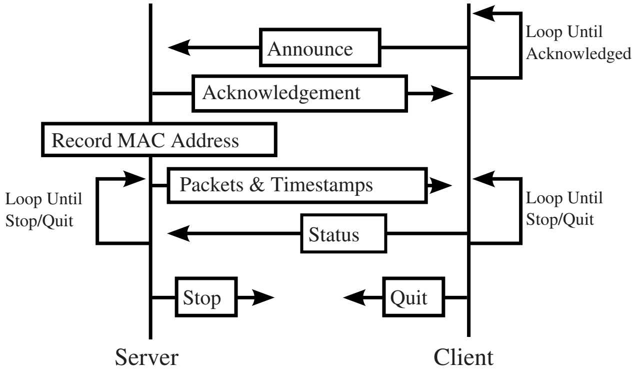


Fig. 5. Illustration of the communication protocol between clients and the server

packet delivery. For management convenience, the client includes a name with the ANN message enabling the server to display a name, rather than simply and I.D. number or MAC address on the management interface.

Upon receiving the ANN_ACK message the client terminates the broadcast ANN message, but continues to send statistical updates periodically via STATUS messages. These status messages enable the system administrator to view logging statistics for each client individually as well as system statistics such as disk usage. Details on the statistical reports and management interface will be detailed shortly.

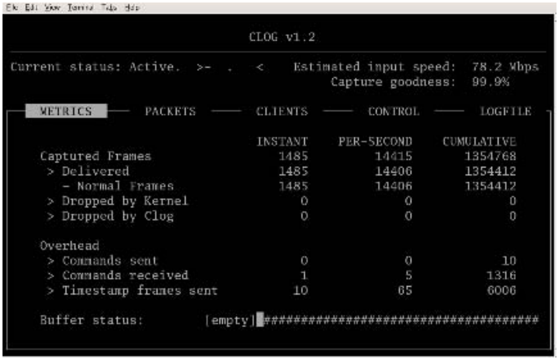
Client systems must quickly check the *Ethertype* field of each Ethernet header in order to effectively process management communication as network tap data arrives on the same data port. When the capture is complete the server can send a STOP message to the client, notifying the client that additional packets will not be delivered. In addition, a client may also send a QUIT message if it must be removed from the system for some reason.

3.4 Management Utility

The management utility is designed to give an administrator the ability to view statistics regarding the data capture and to perform basic functionality such as starting and stopping the capture. A detailed description of each feature of the utility is unnecessary here, but illustration of a few management screens will give a general idea of the reporting system.

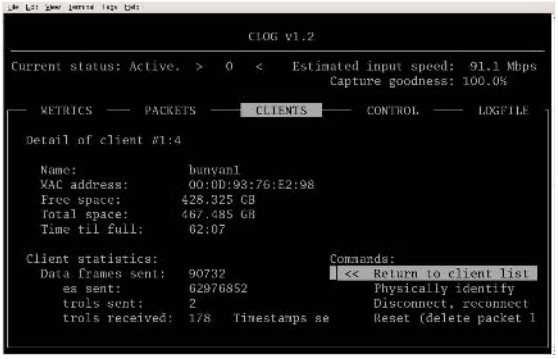
Figure 6 shows the overall statistics of the current capture. The most important feature is the detail of the number of packets dropped, categorized by packets dropped by the system kernel and those dropped by CLog. These statics enable an administrator to quickly identify the existence of a problem if the percentage of dropped packets increases to unexpected levels.

Figure 7 details the statistics of individual systems including total disk usage. This enables an administrator to predict with better clarity when a client may



```
                                 CLOG v1.2
 ---------------------------------------------------------------------------
Current status: Active.  >-   .   <    Estimated input speed:  78.2 Mbps
                                            Capture goodness:  99.9%

 ┌─ METRICS ── PACKETS ─── CLIENTS ─── CONTROL ─── LOGFILE ─┐

                                  INSTANT   PER-SECOND   CUMULATIVE
    Captured Frames                  1485        14415      1354768
     > Delivered                     1485        14406      1354412
       - Normal Frames               1485        14406      1354412
     > Dropped by Kernel                0            0            0
     > Dropped by Clog                  0            0            0

    Overhead
     > Commands sent                    0            0           10
     > Commands received                1            5         1316
     > Timestamp frames sent           10           65         6006

    Buffer status:          [empty]█#####################################
```

Fig. 6. Screenshot of the overall system status report screen



```
                                 CLOG v1.2
 ---------------------------------------------------------------------------
Current status: Active.  >   0   <    Estimated input speed:  91.1 Mbps
                                           Capture goodness: 100.0%

 ┌─ METRICS ── PACKETS ─── CLIENTS ─── CONTROL ─── LOGFILE ─┐
  Detail of client #1:4

    Name:                bunyan1
    MAC address:         00:0D:93:76:E2:98
    Free space:         428.325 GB
    Total space:        467.485 GB
    Time til full:       62:07

  Client statistics:                             Commands:
    Data frames sent:    90732                  █ <<  Return to client list
       es sent:          62976852                     Physically identify
       trols sent:       2                            Disconnect, reconnect
       trols received:   178   Timestamps se          Reset (delete packet 1
```

Fig. 7. Screenshot of the client status report screen

fill up and identify the cause of any problems related to the performance of the data capture.

3.5 Future Enhancements

The CLog system is fully functional and has been used to create high quality network trace files for various research projects. Future enhancements of the system address usability issues of a completed network trace rather than the functionality of the core packet capture process.

Improvements to the management utility include adding the ability to control data which has been collected on the client systems. The current state of the system requires data to be removed manually to other storage areas for experimental use. In addition, in order to create a completely merged trace file a significant amount of storage space is required to consolidate the files. This limits the utilization of the client harddisks if they are required to keep free space for the purpose of creating a merged file.

Currently under development is a feature which allows CLog to control the entire trace without requiring trace files to be merged from each client system.

Following the update of the timestamps for a single client, an index will be created for the trace files on the client. The entire trace for a single client consists of numerous individual files which sequentially numbered with a pre-determined size such as 100MB. The indexing system would enable CLog to quickly locate a position in the entire trace. The goal is to be able to request CLog to replay the network flow based on some parameter such as time-of-day or average bandwidth consumption. An experimental system could then be fed the output from CLog, replaying the feed from a live network.

4 Performance

The performance metrics we address here are directed at the ability of the CLog system to record network traces in terms of the percentage of dropped packets. Processor and memory usage cannot be ignored, but in this case extensive analysis is not necessary due to the minimal impact on the components. CPU usage ranged from 5% to 10% even when processing high bandwidth rates. Memory usage was limited as well and is only an issue when the line speed is greater than the CLog system can process. However, in such a case more memory may not necessarily solve the problem as a larger buffer does not resolve the issue of the buffer being filled faster than it can be emptied.

Performance evaluation was conducted using a large trace file collected from the gateway to the Internet of the University of Notre Dame. The file was replayed using tcpreplay [7] which is able to change the rate of replay to modify bandwidth rates. An actual network trace file is advantageous as the performance of the CLog system is not only affected by the bandwidth of the data, but also by the density in terms of packets per second. Thus, authentic network traffic may give more accurate results than is possible using an artificially generated trace file.

There are two areas of the system where packets may be dropped, in the server or in a client. If there are more packets available than the clients can record then the server is unable to empty its storage buffer quickly enough and packets are lost within the server. Because packets are dropped in this way to handle data overload, client systems are not actually affected by a dramatic increase in bandwidth consumption on the network that is being monitored. However, packet loss in the clients still does occur as show in Figure 8.

The packet loss rate in clients is a function of the number of new log files created. A single trace file was used as input for the experiments which are represented by this graph, meaning the size parameter of the log files was the determining factor on the number of files created. Identical experiments show that tcpdump is not affected in the same manner. Tcpdump was running on the same workstation used as the server for the CLog system utilizing an internal SCSI hard drive for storage. The fact that tcpdump file sizes did not influence the packet loss rate of the capture indicates the CLog clients may be optimized to reduce or eliminate this drawback. This issue will be addressed in future development of the system. For the remainder of the performance measurements log files of size 250MB were used.

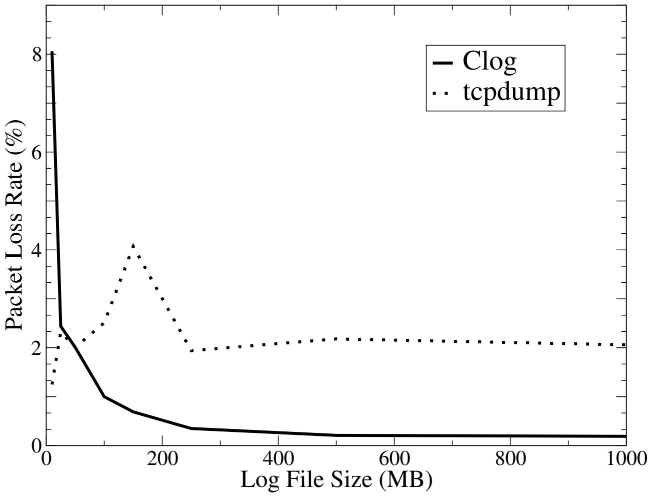


Fig. 8. Illustration of packet drop rate of a client system

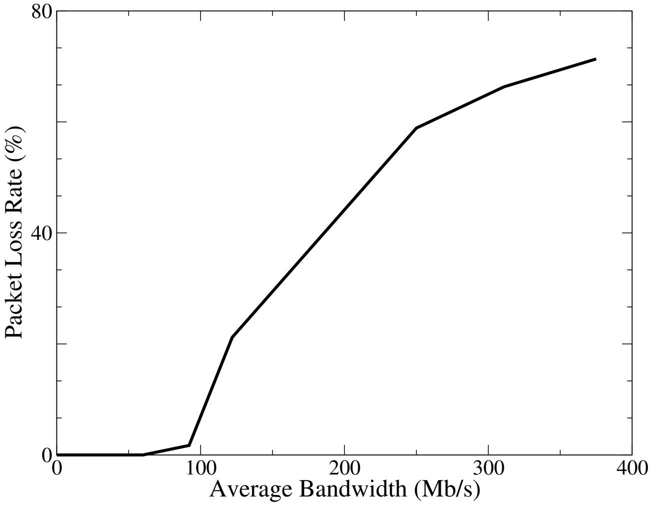


Fig. 9. Performance of a single client Cheap Logger implementation

The overall performance of the system can be determined by measuring the packet loss on the server while varying the number of clients and the speed of the input data. Figure 9 shows the packet loss rate of the system with only a single client logging packets. The input speed is the average speed over the trace file replay. Peak bandwidth during the replay is approximately 50% higher than the average speed.

A single client is able to avoid packet loss at approximately an average bandwidth speed of 85Mb/s. At higher rates the storage buffer of the server reaches

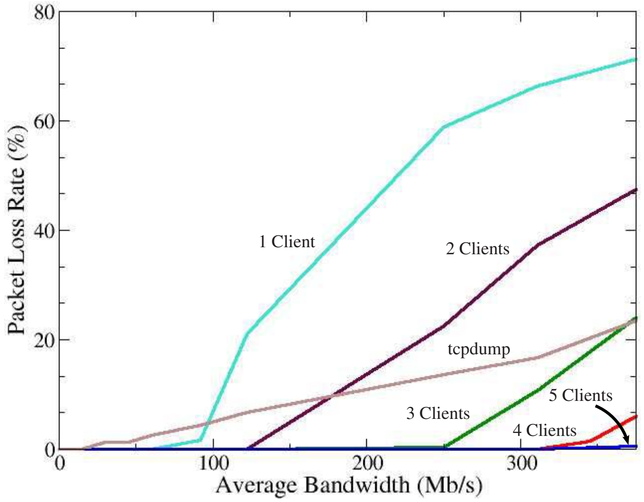


Fig. 10. Comparison of the Cheap Logger system with multiple clients to an implementation of tcpdump

maximum capacity and packets are lost. As the average bandwidth rate increases, the system reaches a threshold where the buffer loses all effectiveness and extreme packet loss occurs. This can be seen in each case where a dramatic increase in packet loss occurs.

Figure 10 illustrates the same data associated with a varying number of clients as well as with the tcpdump packet logger. The original problem the CLog system was designed to overcome was the inability of a hard disk to keep up with the data served by a network monitoring feed. This figure clearly shows the effectiveness of the solution as the introduction of additional client systems greatly improves over the single tcpdump system. It is important to implement a sufficient number of clients for the rate of the data to be recorded as the CLog system reaches a saturation point where packet loss increases dramatically and, in fact, performs much worse than tcpdump. When a sufficient number of clients are added, however, it is possible to record a much more complete trace of network data flow.

At an average data rate of approximately 375Mb/s the number of dropped packets for the five client system is non-zero, although somewhat negligible (0.6%). We note this because the packet loss is reported not as a loss within the CLog system, but as a result of the kernel dropping packets. We have not determined the exact cause of the packet loss, and therefore do not know if this is a result of hardware system capabilities or the CLog system. At an average data rate of approximately 470Mb/s tcpreplay seems to level off and is not able to replay our trace file at greater speeds. At this speed the CLog system still does not report any packet loss, but loss due to kernel packet drops was measured at 0.9%.

5 Summary

Capturing complete network trace files can be very difficult where speeds are high and budgets are low. The *Cheap Logger* (CLog) system is designed to eliminate this tradeoff by providing high quality network data capture without breaking the bank. The system utilizes COTS hardware and is easily scaled with the addition of individual client systems. Once a data capture is complete simple post processing steps reconstruct the original flow by merging files from the distributed logging system. Timestamp synchronization is handled via an efficient timing reporting mechanism and postprocessing of the logged files.

The system significantly outperforms a single system running tcpdump, a very commonly used packet logging application. The current version of the CLog system can be obtained at http://gipse.cse.nd.edu/CLog. Future enhancements to the file creation process of the client application and improvements of data management capabilities are planned for the system.

References

1. Conduant corporation. http://www.conduant.com/.
2. Ethereal. http://www.ethereal.com.
3. Libnet c library. http://www.packetfactory.net/libnet/.
4. Libpcap c library. http://www.tcpdump.org.
5. Network flight recorder. http://www.nfr.net/.
6. Tcpdump. http://www.tcpdump.org.
7. Tcpreplay project. http://tcpreplay.sourceforge.net/.
8. H.-A. Kim and B. Karp. Autograph: Toward automated, distributed worm signature detection. In *Proceedings of USENIX Security Symposium*, pages 271–286, San Diego, CA, August 2004.
9. L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.
10. S. McCanne and S. Floyd. ns Network Simulator. http://www.isi.edu/nsnam/ns/.
11. S. Singh, C. Estan, G. Varghese, and S. Savage. Automated worm fingerprinting. In *Proceedings of USENIX OSDI*, San Francisco, CA, December 2004.

An Efficient Overlay Link Performance Monitoring Technique

Zhi Li[1], Lihua Yuan[2], and Prasant Mohapatra[2]

[1] Network Systems Engineering, AT&T
lizhi@cs.ucdavis.edu
[2] University of California, Davis
{lyuan@ece, prasant@cs}.ucdavis.edu

Abstract. Link performance monitoring is a common task required by different overlay networks. Current overlays typically let each node monitor links by itself, which is not scalable for large networks. Earlier improvement proposals either use a centralized approach or sacrifice measurement accuracy. This paper proposes *MONET*, a distributed overlay monitoring technique. Based on the proposed *X-Set* concept, MONET enables peer cooperation so that each node performs a minimum amount of measurement but can deduce the performance of any link. It does not lose accuracy and adapts to IP-layer path dynamics. Theoretical analysis and simulation results, in terms of monitoring cost and querying overhead, are also discussed in this paper.

1 Introduction

An overlay network is a virtual network formed by a subset of nodes in the underlying layer and virtual links composed of one or more hops on the lower-layer links. Recent research has shown a promising future for using application-layer overlay network to introducing new applications and services, e.g. multicast [1], Quality of Service (QoS), resilient routing, peer-to-peer file sharing, all without disrupting the operation of the lower IP layer. To better support the many upcoming overlay applications, researchers have proposed a generic *overlay service network* (OSN) [2, 3]. An OSN implements common functionalities among application-specific overlays, e.g. overlay link performance monitoring, topology construction, overlay service composition, and provides them as services to applications. It can coordinate the activities of multiple overlays and reduce overhead by avoiding repeating common tasks.

Monitoring overlay link performance, e.g. delay or loss rate, is a common task required by many applications. An overlay network with n nodes might need to monitor n^2 links, with each monitoring job incurring its own overhead. For overlay networks with large number of nodes, a scalable solution is necessary. In fact, several existing random measurement works have already incurred significant amount of overhead to the Internet. Reducing the monitoring overhead while maintaining the measurement accuracy is a challenging task that remains to be fully addressed.

This paper proposes a scalable monitoring service overlay network (MONET), which aims to measure the performance of overlay links and provide timely results to any querier. The key idea is based on the proposed *X-Set* concept (Sec. 3) through which overlay nodes

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 513–524, 2006.

can share measurement information and deduce the performance of some links without directly measuring them. MONET can effectively reduce the monitoring overhead and distribute the measurement load among overlay nodes, all without scarifying accuracy. It also adapts well to lower-layer dynamics like IP path changes.

The rest of this paper is organized as follows. Sec. 2 presents some related work. Sec. 3 discusses *X-set*, which is the foundation of the link selection algorithm used in MONET. Sec. 4 presents the framework and detailed operations of MONET. We present some analysis in Sec. 5, simulation studies in Sec. 6, and conclude in Sec. 7. Due to space limitations, the detailed proof and additional simulation results are referenced for interested readers [4].

2 Related Work

Internet measurement is an active research field. Extensive work has been dedicated to inferring per-link performance when limited information is available [5,6]. In overlay networks, measurement is focused on the performance of overlay links instead of individual IP links. Several works have been done to infer the distance between two arbitrary end hosts [7,8,9]. However, their approaches are only applicable to estimate the approximate end-to-end distances (delay), which is different from our goal of providing accurate overlay link performance information.

Shavit et al. [10] use algebraic tools to compute the link distances that are not directly measured. Given some tracers and some direct path measurement results, the proposed method can infer the performance of some paths or path segments. However, they do not deal with the selection of directly monitored links, sharing monitoring results, or providing scalable monitoring service, which are necessary for overlay networks. Chen et al. [11] also use an algebraic method to show how to use minimal linearly independent k paths to represent the performance of all n^2 paths. Both methods and MONET try to exploit the IP-layer information for reducing the overlay link monitoring overhead. The approach in [11] uses a centralized approach to determine directly monitored overlay links. In contrast, MONET proposes a distributed approach and tradeoff overhead reduction for scalability to large overlay network. In addition, MONET can cope with dynamic IP-layer path changes and avoid single point failure.

Tang et al. also propose approaches to reduce the number of directly monitored overlay links and track all the performance of all possible overlay links [12]. It has a centralized version and a distributed version. Their approaches are based on the assumption that an overlay link performance is approximately similar to the performance of its sub-segments, which can not provide accurate monitoring results.

3 X-Set

Although an overlay node can measure the performance to any other nodes, it is possible for overlay nodes to share information and reduce measurement overhead if there is sufficient knowledge about the IP topology. Fig. 1 depicts a simple scenario in which node C is on the path of AB. Although there are 3 overlay links, it is sufficient to monitor the delay of any two of them and then deduce the other. For an *additive* metric

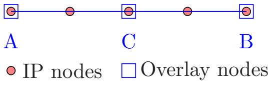


Fig. 1. On-Path Overlay Nodes

$$\overline{AB} = \overline{AC} + \overline{CB} \quad (1)$$

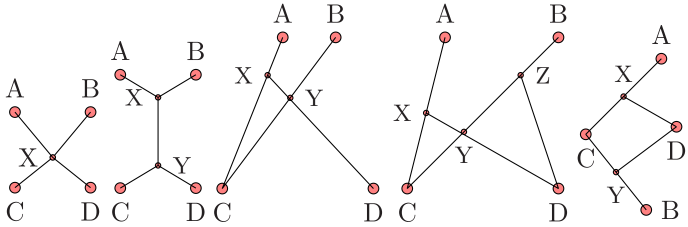
$$\widetilde{AB} = 1 - (1 - \widetilde{AC})\,(1 - \widetilde{CB}) \quad (2)$$

$$log(1 - \widetilde{AB}) = log(1 - \widetilde{AC}) + log(1 - \widetilde{CB}) \quad (3)$$

like delay, the relationship of the three links can be expressed with Eq. 1. Similarly, the loss rate of link AB ($\widetilde{AB}$), which is a *multiplicative* metric, can be found using Eq. 2. This paper focuses on additive metrics since a multiplicative metric can be transformed to additive metric at log-scale (Eq. 3).

For an overlay node A, if we map its paths to all other nodes onto the underlying IP topology, all the IP paths form a *source-based routing tree* (SRT) rooted as this node. Similarly, the IP paths from other nodes to A form a *destination-based routing tree* (DRT). Both source and destination-based routing trees can be decomposed into a set of basic components in the shape of a reverse “Y”. We can use this basic component to reduce the number of overlay links we need to monitor. Consider a simple topology with four overlay nodes A, B, C and D. A and B need to monitor the performance of the overlay links (AC, AD, BC, and BD) connecting to nodes C and D. Based on the SRT of A or B, the two paths to C and D (from A to C, D and from B to C, D respectively) can be decomposed into two "Y"s. The different combinations of the two “Y”s are shown in Fig. 2, in which, X, Y and Z are non-overlay, on-path nodes. Note that Fig. 2 does not include every possible combination of two “Y”s. However, any other scenario can be reduced to one of the graphs in Fig. 2.

Equations in Table 1 present the relationship among the performance of overlay links for topologies illustrated in Fig. 2. For an additive metric, the performance of an



(a) **(b)** **(c)** **(d)** **(e)**

Fig. 2. Different Combinations of Two "Y"s

Table 1. Mathematical Expression of the Graphs in Fig. 2

Path	$\overline{AC}$	$\overline{AD}$	$\overline{BC}$	$\overline{BD}$
(a)	$\overline{AX} + \overline{XC}$	$\overline{AX} + \overline{XD}$	$\overline{BX} + \overline{XC}$	$\overline{BX} + \overline{XD}$
(b)	$\overline{AX} + \overline{XY} + \overline{YC}$	$\overline{AX} + \overline{XY} + \overline{YD}$	$\overline{BX} + \overline{XY} + \overline{YC}$	$\overline{BX} + \overline{XY} + \overline{YD}$
(c)	$\overline{AX} + \overline{XC}$	$\overline{AX} + \overline{XY} + \overline{YD}$	$\overline{BY} + \overline{YC}$	$\overline{BY} + \overline{YD}$
(d)	$\overline{AX} + \overline{XC}$	$\overline{AX} + \overline{XY} + \overline{YD}$	$\overline{BZ} + \overline{ZY} + \overline{YC}$	$\overline{BZ} + \overline{ZD}$
(e)	$\overline{AX} + \overline{XC}$	$\overline{AX} + \overline{XD}$	$\overline{BY} + \overline{YC}$	$\overline{BY} + \overline{YD}$

overlay link is the combination of the performance of all its sub-segments. Using a similar theory to that used by Chen et al. [11], if there are linearly dependent equations within a set of overlay link performance expressions, some of the overlay links can be removed from the set of links to directly measure without affecting the accuracy. The total number of overlay links these two nodes (A and B) need to directly monitor is the rank of this set of equations. The corresponding directly monitored overlay links are the linearly independent equations. For the equation sets in Fig. 2, it is easy to see that the equations in sets (a) and (b) can be decomposed into three equations, which means that the performance of the four overlay links in Fig. 2a and Fig. 2b can be obtained by directly monitoring three overlay links. For example, if node A monitors the performance of AD, node B monitors the performance of BC and BD, A can obtain the performance of AC since $\overline{AC} = \overline{AD} + \overline{BC} - \overline{BD}$.

Definition 1. ***X-Set****: For two overlay nodes, if their IP layer paths to the other two overlay nodes can be reduced to Fig. 2a or Fig. 2b, the four overlay links form an X-Set. The performance of all the four overlay links can be obtained by directly monitoring any three of them.*

The basic requirements for two Ys to compose an *X-Set* is that the two branching nodes of the two "Y"s overlap with each other such as node X in Fig. 2a and Y in Fig. 2b. Based on this, two nodes can cooperate with each other to find *X-Set*s (the details of which are described in the next section). As "Y"s are the basic components of SRTs, finding X-Sets is the basic method for two overlay nodes to cooperate and reduce the total number of directly monitored overlay links. More complicated combinations of the SRTs of two nodes can be partitioned into multiple X-Sets, which then allows the total number of directly monitored overlay links to be reduced.

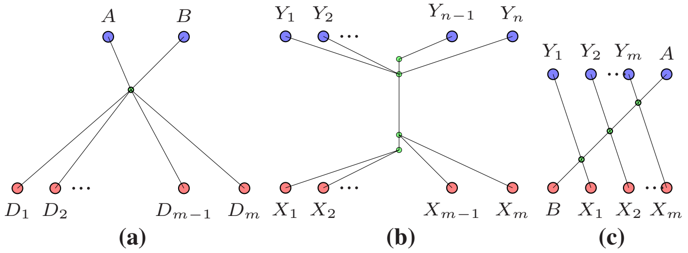


Fig. 3. Combinations of X-Sets

For example, in Fig. 3a, overlay node A and B need to track the performance of the $2m$ incident overlay links to the destination nodes (from D_1 to D_m). The graph can be seen as the combination of $m - 1$ X-Sets ($\{A, B, D_1, D_2\}, \{A, B, D_1, D_3\}, \cdots \{A, B, D_1, D_m\}$). For the first X-Set, we only need to directly monitor three overlay links. For each of the remaining $m - 2$ X-Sets, one only needs to directly monitor one additional link to obtain the performance of two links (as we already have the performance of AD_1 and BD_1). The total number of directly monitored links is $m + 1$ instead of $2m$.

4 A Framework of Monitoring Service Overlay Network (MONET)

MONET assumes that an overlay link observes many more performance (delay or loss rate) changes than underlying IP path. An overlay node can identify its IP routing information to other nodes, by either querying other service modules or by *traceroute*. In MONET, overlay nodes independently determine their incident overlay links and continuously monitors their performance. They also share these monitoring results with a set of neighbors.

Given an IP topology $G(V, E)$ and a set of overlay nodes $V' \in V$, each overlay node independently chooses its set of directly monitored overlay links either based on its local information or by collaborating with other overlay nodes. These overlay links form the topology $G'(V', E')$ of the MONET, in which each link in G' is an IP path in G. Based on the MONET topology, each overlay node continuously monitors the performance of its incident overlay links. The links in the MONET topology are *directly monitored*. Other links in the corresponding full-mesh topology are called *indirectly monitored overlay links*, whose performance can be derived by the directly monitored results. In other words, MONET aims to track the performance of n^2 overlay links but incurs the least amount of monitoring overhead. Meantime, MONET aims to minimize the communication overhead and balance the load among the overlay nodes.

4.1 How Does MONET Work?

In MONET, each overlay node maintains an *overlay monitoring table* (OMT), which is an essential component to provide overlay link monitoring services. One entry is created for each adjacent overlay node in the full mesh topology. Each OMT entry has three fields: $DestID, MonitorBool, MethodList$. $DestID$ is the address of the neighbor – the destination of this overlay link. $MonitorBool$ determines whether the current overlay node (the OMT owner) should directly monitor the performance of the corresponding overlay link or not. If not, the $MethodList$ field includes the list of methods to obtain the performance of this link and the maximal query hops for each of these methods. For each indirectly monitored overlay link, an overlay node may have more than one method to obtain the corresponding overlay link performance. In our simulation studies (Sec. 6), we assume that each node only maintains one method for each indirectly monitored overlay link.

For example, in Fig. 1 and Fig. 2a, A can use one of the two methods to obtain the performance of link AC: $\overline{AB} - \overline{BC}$ or $\overline{AD} + \overline{BC} - \overline{BD}$. A needs to query one hop for the performance of BC. When a query arrives at node A for link AC performance, A first locates the corresponding entry for AC from its OMT. If the entry's $MonitorBool$ is true, it can directly return the overlay link performance. Otherwise, it will obtain the methods from MethodList and try each of them to obtain the link performance. Based on method $\overline{AB} + \overline{BC}$, besides checking the entry for AB, node A also needs to send a query to B for the performance of link BC, or, based on $\overline{AD} + \overline{BC} - \overline{BD}$, it will send a query for $\overline{BC}$ and $\overline{BD}$. If any performance query request returns, A can obtain the performance of overlay link AC. It is easy to see that a link performance query may take several query hops to return the performance. To balance the tradeoff between the query distance and query overhead, a node can try each of the methods (or a subset of them) in parallel or sequentially.

4.2 Find Directly Monitored Overlay Links

Besides an OMT, each overlay node also needs to maintain two other data structures: a list of *Friend Nodes* and the corresponding list of "Y"s for each Friend Node. To fill each OMT entry, the overlay nodes can take the following two steps.

Algorithm 1. Finding "Y"s

```
$Y - Set \leftarrow \emptyset$ //Initialization
for each overlay node $X$ do
  Retrieve the paths to every other nodes
  Construct the SRT rooted at $X$
  for each overlay nodes pair $A$, $B$ do
    Find the branching node $BN_{AB}$ // the
      furthest common node of XA and XB
      in the SRT
    if $BN_{AB} \neq X$ do
      Append $< A, B, BN_{AB} >$ into Y-Set
```

Algorithm 2. Load Balancing

```
Input: X-Set (A, B, X, Y), $S \leftarrow$ size of overlay
Require: $ID_A$<$ID_B$ and $ID_X < ID_Y$
if $ID_Y < S * 1/(2^{1/2})$
  if $ID_B < S * 1/(2^{1/2})$ [case 1]
    $A \rightarrow \{AY, AX\}$, $B \rightarrow \{BX\}$
  else    $A \rightarrow \{AX\}$, $B \rightarrow \{BY, BX\}$ [case 2]
else
  if $ID_B < S * 1/(2^{1/2})$ [case 3]
    $A \rightarrow \{AY, AX\}$, $B \rightarrow \{BY\}$
  else    $A \rightarrow \{AY\}$, $B \rightarrow \{BY, BX\}$ [case 4]
```

First, each node independently identifies its list of "Y"s using Algo. 1. The main idea of Algo. 1 is to construct the SRT so that a node can locate the branching nodes and "Y"s. Based on the IP paths to other overlay nodes, a node can also find the possible scenarios as described in Eq.1. In addition, a node can also choose a set of overlay nodes as *friend nodes*, with which the node will share monitoring results. The selection of friend nodes is based on the IP path distance because the closer the two nodes are, the higher the chance that their incident overlay links will compose X-Sets.

In the second step, an overlay node will exchange its list of "Y"s information with the selected Friend Nodes. Based on the "Y" information from its friend nodes, a node can identify the *X-Sets* by comparing the common branching nodes of the two "Y"s for any two destination nodes. To balance the overhead from directly monitoring among the overlay nodes and to avoid the complicated negotiation procedure, an overlay node uses Algo. 2 to select its directly monitored overlay links.

The input to Algo. 2 is an X-Set with source nodes as A, B and destination nodes as C, D. The main idea is based on the ID values of the four nodes, which is a unique number defined by either IP address or MAC address. The probability of $(ID_X < ID_Y)$, $(ID_Y < S * 1/(2^{1/2})$, $(ID_A < ID_B)$ and $(ID_B < S * 1/(2^{1/2})$ are all $1/2$. Considering both case 1 and case 3, for the probability of $1/2$, node A only needs to monitor one overlay link for an X-Set. We can conclude that this algorithm balances the monitoring overhead (both sending and receiving measurement probing traffic) among the different overlay nodes without complicated negotiation procedures.

4.3 Dealing with Dynamic Network Condition

In MONET, each node periodically (much less frequently than link performance probing) performs traceroute or other methods to obtain the IP-layer path information. If a node realizes that an IP-layer path to the other node changes, it will check whether there is any change in its set of "Y"s. If necessary, it will update some overlay links' monitoring methods. In addition, it will also send the "Y" update information to its friend nodes, which in turn may need to update their OMTs. The procedure of updating OMT entries is similar as adding OMT entries as discussed above.

Similarly, when an overlay node joins an existing overlay network, it first retrieves the IP-path information to other overlay nodes and chooses its friend nodes. After finding its "Y"s in its SRT and receiving "Y" information from its friend nodes, it can begin to find "X-Set" and fill its OMT entries one-by-one. If an overlay node needs to update its friend nodes set, it can also take similar steps.

In some cases, overlay nodes may not be able to retrieve the complete IP path information. For example, an IP path traceroute result could be like "69.110.237.117, *, 171.66.1.17, 171.67.255.249, *, *, 171.66.7.234". As the *X-Set* technique in MONET is based on the overlapping of two "Y"s' branching nodes, the incomplete path information will only affect the number of "Y"s each overlay node can find. It may result in the increase in the number of directly monitored overlay links. However, it will not affect the correctness and normal operations of MONET.

In summary, each MONET node independently (by exchanging information with a selected set of friend nodes) chooses which methods are used to track overlay link performance by either direct monitoring or indirect monitoring. Using the proposed techniques, MONET can effectively reduce the number (cost) of directly monitored overlay links without affecting the monitoring accuracy. In addition, it can quickly handle IP topology or IP path changes and dynamic overlay network membership.

5 Performance Analysis

5.1 Number of Overlay Links in MONET Topology

Fig. 3b and Fig. 3c shows the two different combinations of *X-Set*s. In Fig. 3b, node X_1 needs to monitor the performance of links from itself to node $Y_1, \cdots, Y_n$. Suppose X_1 realizes that the path of other $m-1$ nodes $(X_1, X_2, \cdots, X_m)$ to $Y_1, Y_2, \cdots, Y_n$ compose multiple X-Sets as shown in the graph. We can estimate the average number of links a node X_1 need to directly monitor in order to obtain the performance of all its links $(X_1Y_1, X_1Y_2, \cdots, X_1Y_n)$ based on the following analysis. First, X_1, X_2, Y_1 and Y_2 compose the first X-Set; the total number of directly monitored overlay links (for both X_1 and X_2) is 3. After this, if X_1 and X_2 want to monitor the links to one additional node (such as the links to Y_3, X_1Y_3 and X_2Y_3), they only need to directly monitor one more link to obtain the performance of two (X_1Y_3 or X_2Y_3). If another node (such as X_3) wants to monitor the performance to Y_1 and Y_2 (X_3Y_1 and X_3Y_2), it only needs to directly monitor one additional link (X_3Y_1 or X_3Y_2). Consequently, in order for all the nodes $(X_1, X_2, \cdots, X_m)$ to obtain the overlay links' performance to all the destination nodes $(Y_1, Y_2, \cdots, Y_n)$, the total number links $X_1 \cdots X_m$ need to directly monitor is $m+n-1$. On average, for each overlay link, one node needs to have $\frac{1}{m} + \frac{1}{n} - \frac{1}{mn}$ incident links in the MONET topology (average per node and per link directly monitoring cost) to obtain the performance of all the links.

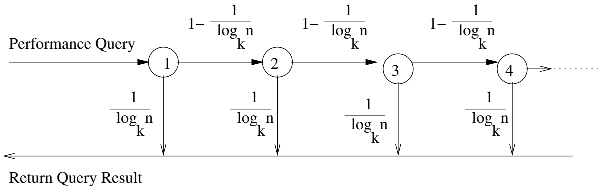


Fig. 4. Link Performance Query Processing Steps

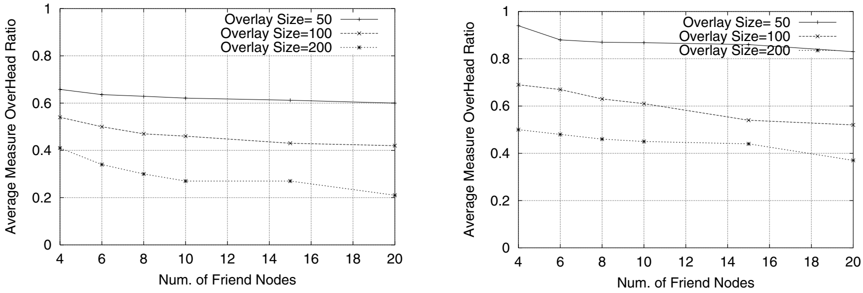
As mentioned above, the routing path of an overlay node to other nodes can be mapped to a SRT rooted at itself. For a connected graph, all the overlay nodes are located at the leaf nodes of other nodes' SRTs. Assume the total of n nodes are in the MONET and the average branching degree in the routing tree is k. The average height of the tree is h ($h = \log_k^n$). For a routing tree, it has different levels of sub-trees: the level 0 sub-tree is itself; level 1 sub-trees are the sub-trees that rooted at the children nodes of the root;...;level h sub-trees are the leaf nodes.

5.2 Overlay Link Performance Query Hops

A node of MONET does not monitor all adjacent overlay links directly. Therefore, it may need to query other nodes, which may repeat the similar process, to infer the performance of the link under request. We use *Link Performance Query Hops* to evaluate the average query distance to fulfill each overlay link performance query. As depicted in Fig. 4, the average directly monitoring cost of each overlay link is $\frac{1}{\log_k^n}$. The number of incident overlay links for each node is $n * \frac{1}{\log_k^n}$. A link performance query processing procedure is shown in Fig. 4. Suppose a query arrives at node 1. Node 1 has a probability of $\frac{1}{\log_k^n}$ to respond to the query without querying others. Otherwise, it will forward the request to the next node (e.g. node 2) based on its OMT. Node 2 will then repeat the same procedure: either returns the query result to node 1 with probability of $\frac{1}{\log_k^n}$ or sends another query based on its OMT to node 3. Consequently, the average query hops can be found as $\log_k^n - 1$. One can show that the upper bound of the directly monitoring cost for each overlay link is $\frac{1}{\log_k^n} - \frac{k}{\log_k^n * n}$. The cost is inversely proportional to the average height of the SRTs. Given a fixed number of overlay nodes, the smaller value of the average degree is in the routing tree, the lower monitoring cost each overlay link incurs (less directly monitored overlay links in the MONET topology).

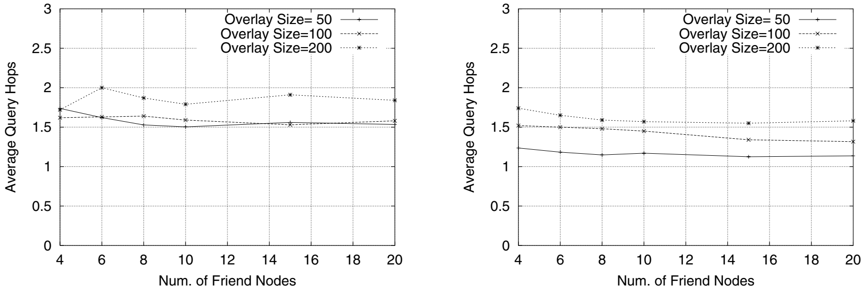
6 Simulation Study and Discussions

We evaluate the performance of MONET through simulation. The simulations are based on a real ISP intra-domain topology (Intra604) taken from Rocketfuel [13] and three topologies generated by BRITE [14]. *Intra604* has 604 nodes, 4547 directed links and an average node degree of 7.5. For the other three topologies, *W1000* is a router-level Waxman [15] topology with 1000 nodes. The other two (*H1000* and *H5000*) are two 2-layer hierarchical topologies with the lower level based on Waxman model and the higher level based on Barabasi-Albert model, with 1000 and 5000 nodes respectively.



(a). Intra604 **(b).** H1000

Fig. 5. Monitoring Overhead vs. Num. of Friend Nodes



(a). Intra604 **(b).** H1000

Fig. 6. Average Indirectly-Monitored Overlay Link Performance Query Hops

For the topologies generated by BRITE, each node is adjacent on average of two undirected links, which leads to an average node degree of 4.0. The IP layer use shortest path based routing. We focus on the following performance metrics: average query hops for indirect monitored overlay links, average overlay link monitoring overhead, monitoring overhead balancing results, and OMT table updates under dynamic IP-layer change. Due to space limitation, we only present the results on *Intra604* and *H1000*.

6.1 Monitoring Overhead

We use *Monitoring Overhead Ratio* (MOR) to evaluate the performance of MONET in reducing the monitoring overhead of each node to provide constant link performance monitoring service. For an overlay node, MOR is defined as:

$$MOR = \frac{\text{\# of Adjacent Directly Monitored Overlay Links}}{\text{Overlay Network Size}} \tag{4}$$

The directly monitored overlay link means that the overlay node keeps sending probing traffic to monitor the overlay link performance. It is easy to see that the smaller value of MOR means that MONET can provide better performance in terms of decreasing the monitoring overhead. Fig. 5 shows that the average MOR for overlay networks of various sizes and numbers of friend nodes on top of the different IP topologies. From the simulation results, one can observe that the increase in the number of friend nodes will reduce the average MOR, which means the monitoring overhead will decrease. This is because each node has higher chance to find X-Sets with its neighbors. However, as we mentioned above, the "Y" information needs to be shared between friend nodes. The larger number of friend nodes means that higher amounts traffic will be exchanged during dynamic IP-layer path changes. When considering different sizes of overlay networks on the same underlying IP topology, we can observe that the performance of MONET varies greatly. The larger an overlay network is, the less the average MOR is. This is because each node can have more candidate nodes to choose as friend nodes. This will result in more X-Sets, which means that the number of directly monitored overlay links can potentially decrease.

6.2 Average Query Hops

An overlay node needs to query others when a query for an indirectly monitored overlay link arrives and then relay the answer back. The number of query hops determines the response delay and the accuracy of the performance value. Fig. 6 shows that the average number of query hops for all indirectly monitored links. We can observe that the average query hops are all below 2.0 for the various simulated scenarios. The average query hops in Fig. 6a (between 1.5 and 2.0) is higher than Fig. 6b (between 1.0 and 1.5). This is because the first topology is smaller, resulting in a higher probability of finding X-Sets. Considering together with the simulation results of the average MOR and average query hops, we can conclude that lower MOR is correlated to longer query hops which agrees with our previous analysis results. In addition, we can observe that increasing the number of friend nodes only slightly affects the average query hops.

6.3 Balancing the Monitoring Overhead

MONET aims to balance the overlay link monitoring overhead among all the overlay nodes as described in Algo. 2. For comparison, we consider another none-load-balancing monitoring overhead distribution method: if A and B (assume $ID_A < ID_B$) find the overlay links connecting to X and Y (assume $ID_X < ID_Y$) form an *X-Set*, A will never monitor link AX but always deduce its performance based on the values of the other three links. Note that both methods allows overlay nodes to share the monitoring overhead and collaborate without complicated negotiation and message exchanges.

Fig. 7 presents the effect of load balancing based on *Intra604* with 4 friend nodes. The x-axis values are the different bins of MOR values while the y-axis shows the numbers of overlay nodes within the corresponding bins. With load balancing, more nodes are located in the bins whose value are closer to the average MOR. This suggests that Algo. 2 can effectively distribute the overhead among nodes. In contrast, if load balancing is not available, some nodes will have high monitoring overhead while others are lightly loaded.

6.4 Effect of IP-Layer Path Change

The operation of MONET is based on the IP-layer path information. If there is an IP-layer path change in the overlay links, some X-Sets will be added or deleted, which

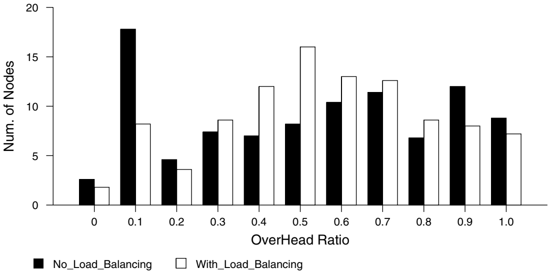


Fig. 7. Distribution of Monitoring Overhead Ratio

Table 2. The Effect of IP-layer Path Changes (Overlays on Top of Intra604 Topology)

Overlay Size	# of Friends	IP-layer path Failure Ratio	# of IP Link Failures	# of Overlay Path Changes	# of "Y" Changes	# of OMT Updates
50	4	0.001	6	14.66	3.75	5.6
50	8	0.001	6	15.64	4.70	3.43
50	10	0.001	6	13.54	5.21	3.45
50	15	0.001	6	10.18	4.82	5.30
50	4	0.002	11	24.76	6.86	4.48
50	8	0.002	11	23.68	7.70	5.05
50	10	0.002	11	25.30	9.5	7.0
50	15	0.002	11	31.90	8.6	10.03
100	4	0.001	6	47.75	9.2	6.14
100	8	0.001	6	45.10	9.58	20.35
100	10	0.001	6	50.0	10.38	29.4
100	15	0.001	6	51.38	13.3	40.92
100	4	0.002	11	122.76	18.5	14.96
100	8	0.002	11	113.79	19.12	25.09
100	10	0.002	11	124.63	27.13	54.43
100	15	0.002	11	125.00	22.73	70.13

leads to the updates of OMT table. In this paper, we use *Intra604* as an example to investigate the effect of IP-layer path changes. We first randomly form an overlay network with size 50 or 100. After this, each node runs the MONET to set up its OMT table. After the system stabilizes, we randomly fail some IP-layer links without losing the IP-layer connectivity. After this, the affected OMT entries will the refilled by MONET. The relationships between IP-layer, overlay layer, number of "Y" changes as well as the OMT table updates are shown in Table 2. From the results, we can observe that the increase of the IP-layer link failure ratio will increase the number of changes in the overlay links' IP-layer paths. This is because that whenever the average number of friend nodes is increased, the affected number of X-Sets (added or deleted) also will be increased. This will result in larger number of OMT table updates. The OMT updates include the changes between direct overlay link monitoring and indirect monitoring, as well as the changes between different indirect monitoring methods. However, even under higher IP-layer path failure ratios (0.001 or 0.002), the average number of OMT updates is very small, less than 0.5 entry per node. This is because that even if there are a lot of overlay link IP-layer paths change, the new paths will most likely take similar paths to bypass the failed links. Consequently, even though the locations of X-Set branching nodes change, the composition and the number of X-Sets will more or less remain stable.

7 Conclusion

This paper proposed a framework called MONET to efficiently monitor and provide accurate overlay link performance information. The important mission of MONET is to reduce the monitoring cost while maintaining monitoring accuracy. MONET uses a distributed approach that can evenly distribute the path monitoring overhead and easily deal with IP-layer path changes. We also presented some analysis and simulation results in terms of monitoring overhead reduction, link performance query hops and monitoring load balancing.

References

1. Chu, Y.H., Rao, S.G., Zhang, H.: A case for end system multicast. In: Measurement and Modeling of Computer Systems. (2000)
2. Lakshminarayanan, K., Stoica, I., Shenker, S.: Building a flexible and efficient routing infrastructure: Need and challenges. Technical report, UC Berkeley UCB/CSD-03-1254 (2003)
3. Braynard, R., Kostic, D., Rodriguez, A., Chase, J., Vahdat, A.: Opus: an overlay peer utility service. In: IEEE OpenArch'02. (2002)
4. Li, Z., Yuan, L., Mohapatra, P.: An efficient overlay link performance monitoring technique. Technical Report CSE-2005-28, Computer Science, University of California, Davis (2005)
5. Padmanabhan, V.N., Qiu, L., Wang, H.J.: Server-based inference of interent performance. In: Proc. IEEE INFOCOM. (2003)
6. Caceres, R., Duffield, N., Horowitz, J., Towsley, D.: Multicast-based inference of network-internal loss characteristics. In: Proc. IEEE INFOCOM. (1998)
7. Ng, E., Zhang, H.: Predicting internet network distance with coordiantes-based approaches. In: IEEE INFOCOM. (2002)
8. Tang, L., Crovella, M.: Virtual landmarks for the Internet. In: ACM SIGCOMM/USENIX IMC. (2003)
9. Dabek, F., Cox, R., Kaahoek, F., Morris, R.: Vivaldi: A decentralized network coordinate system. In: ACM SIGCOMM. (2004)
10. Shavitt, Y., Sun, X., Wool, A., Yener, B.: Computing the unmeasured: An algebraic approach to internet mapping. In: Proc. IEEE INFOCOM. (2001)
11. Chen, Y., Bindel, D., Katz, R.H.: An algebraic approach to practical and scalable overlay network monitoring. In: ACM SIGCOMM. (2004)
12. Tang, C., McKinley, P.K.: On the cost-quality tradeoff in topology-aware overlay path probing. In: ICNP. (2003)
13. Spring, N., Mahajan, R., Wetherall, D.: Measuring isp topologies with rocketfuel. In: Proc. ACM SIGCOMM. (2002)
14. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE. `http://www.cs.bu.edu/brite/`(2002)
15. Waxman, B.M.: Routing of Multipoint Connections. IEEE JSAC (1988)

Measurement of Radio Propagation Path Loss over the Sea for Wireless Multimedia

Dong You Choi

Division of Electronics & Information Engineering, Cheongju University,
#36 Naedok-dong, Sangdang-gu, Cheongju-city 360-764, Korea
dy_choi@cju.ac.kr

Abstract. In order to estimate the signal parameters accurately for wireless multimedia services, it is necessary to estimate a system's propagation characteristics through a medium. Propagation analysis provides a good initial estimate of the signal characteristics. The ability to accurately predict radio propagation behavior for wireless multimedia services is becoming crucial to system design. Since site measurements are costly, propagation models have been developed as a suitable, low cost, and convenient alternative [1]. A number of studies have been conducted to quantitatively predict the characteristics of propagation in inhabited areas on land having many wireless multimedia service users, resulting in a number propagation prediction models being proposed. However, since very few such studies have been conducted for the sea, which has a different physical layer structure from land, the propagation prediction model for free space has been commonly used. Thus, in this study, I measured the propagation path loss of a 1950 MHz band signal over the sea surface, and analyzed the results by comparing them with the path loss data of a propagation prediction model in free space, which is frequently used to predict the propagation path loss over the sea surface.

1 Introduction

The commercial success of wireless communication, since its initial implementation in the early 1980s, has led to there being an intense interest among wireless engineers in understanding and predicting the radio propagation characteristics in various urban and suburban areas, and even within buildings. Given that the explosive growth of wireless multimedia service is continuing unabated, it would be very useful to have the capability of determining the optimum base-station location, obtaining suitable data rates, and estimating their coverage, without having to conduct extensive propagation measurements, which are very expensive and time consuming [1].

Whereas many studies have been conducted to predict the characteristics of propagation quantitatively land, including the development of many propagation prediction models, few such efforts have been conducted for the sea. In fact, there are many difficulties involved in providing wireless multimedia services over the sea, viz. the lack of economic viability associated with long and short distance services, the absence of good locations for new base-stations, and the difficulties associated with these locations. To solve these problems, facility investment and maintenance

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 525 – 532, 2006.

expenses need to be reduced by optimizing the service area per base-station the precise prediction of the propagation path loss over the sea surface.

Accordingly, in this study, I measured the propagation path loss of a 1950 MHz band signal over the sea surface, and analyzed the results by comparing them with the predicted propagation path loss in free space, which is frequently used to predict the propagation path loss over the sea surface.

2 Propagation Environment and Propagation Path Loss

The radio propagation over the sea surface is different from the land propagation prediction model. In other words, the total received power of a mobile unit situated over the sea is the sum of the direct wave, the reflected wave from the sea surface, and the reflected wave from the ground. As a result, it gives more intense interference to other base-stations and mobile units, as compared with land propagation, and so special attention and care is needed. The received power over the sea surface is given by Eq. (1) [2].

$$\begin{aligned} P_r &= P_t \times \left(\frac{\lambda}{4\pi d}\right)^2 \left|1 - e^{jd_{\theta 1}} - e^{jd_{\theta 2}}\right|^2 \\ &= P_t \times \left(\frac{\lambda}{4\pi d}\right)^2 \left|1 - (\cos\ d_{\theta 1} + \cos\ d_{\theta 2}) - j(\sin\ d_{\theta 1} + \sin\ d_{\theta 2})\right|^2 \end{aligned} \tag{1}$$

where d is the path length, λ is the wavelength, $d_{\theta 1}$ is the difference in the propagation path between the direct wave and the reflected wave from the ground, $d_{\theta 2}$ is the difference in the propagation path between the direct wave and the reflected wave from the sea surface, p_t is the transmitting power, and p_r is the received power in free space.

However, in Eq. (1), the difference in the propagation path between the two reflected waves, $d_{\theta 1}$ and $d_{\theta 2}$, is sufficiently small for propagation path loss over the sea surface to be replaced by the predicted value of the propagation path loss in free space, if a limiting value, 0, is adopted in both $d_{\theta 1}$ and $d_{\theta 2}$, as shown in Eq. (2).

$$\begin{aligned} P_r &\approx \lim_{d_{\theta 1}, d_{\theta 2} \to 0} \left[\left\{ P_t \times \left(\frac{\lambda}{4\pi d}\right)^2 \right\} \left|1 - e^{jd_{\theta 1}} - e^{jd_{\theta 2}}\right|^2 \right] \\ &\approx P_t \times \left(\frac{\lambda}{4\pi d}\right)^2 \end{aligned} \tag{2}$$

In general, the propagation path loss in a downtown environment is known to be about 20 ~ 50dB/dec, based on empirical measurements, depending on the environment, and an approximate value of 20dB/dec is generally use for the propagation path loss over the sea surface [1, 3, 4].

3 Experimental Setup and Measurement Procedure

The propagation path loss over the sea surface was measured in the vicinity of the islands situated off the coast of in Latin America.

The transmitting signal (1950 MHz) was generated using a signal generator installed in the steel tower of an existing base-station in Latin America.

The signal sent from the base-station was processed on a real-time basis using HP RF Coverage Measurement equipment manufactured by Agilent. The location information was obtained using a GPS (Global Positioning System) embedded in the receiving set and Mercator projection [4, 5]. To measure the propagation path loss over the sea surface, a boat with a speed of 40 ~ 60km/h was used.

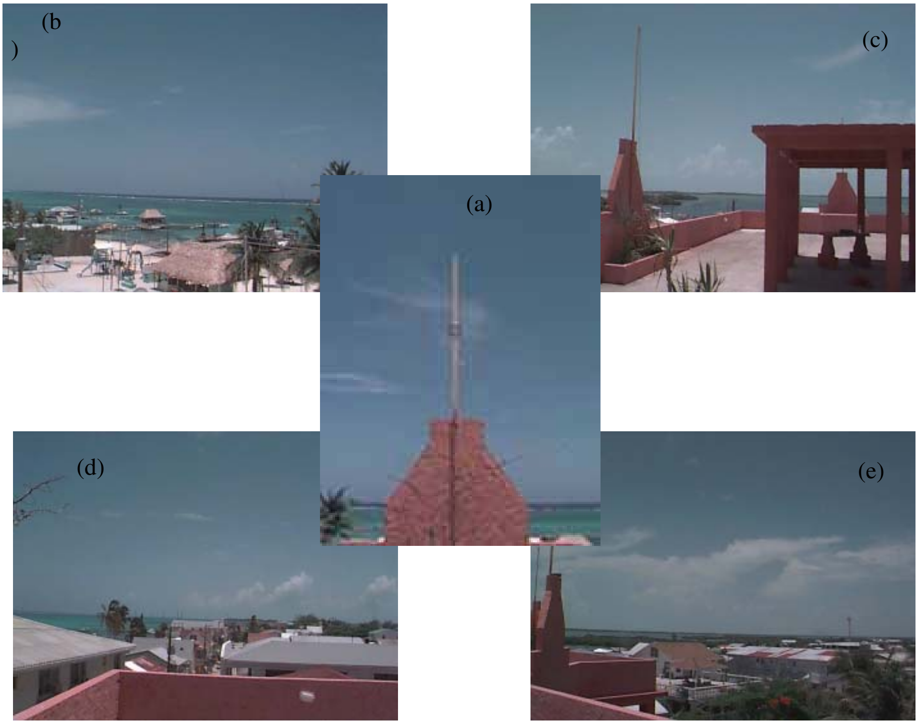


Fig. 1. Sample areal photograph. (a) Test antenna (b) East (c) West (d) South (e) North.

Table 1. Location information

	Latitude	Longitude
Test site	17-55-9.7	(-)87-57-40.6
UTM coordinate	1981526.786	398186.5393

Table 2. Measured parameters

Tx power	Cable loss	Tx ant. gain	Tx ant. height	Rx ant. height
20W (43dBm)	2dB	6dB	22m (Ant. 2m + Building 20m)	2.5m

4 Results

Fig. 2 shows the path loss slope from the measured data. The measured data in Fig. 2 represent the mean values of tens ~ hundreds of data collected when the mobile unit's location was changed by 0.001 degrees in latitude or longitude.

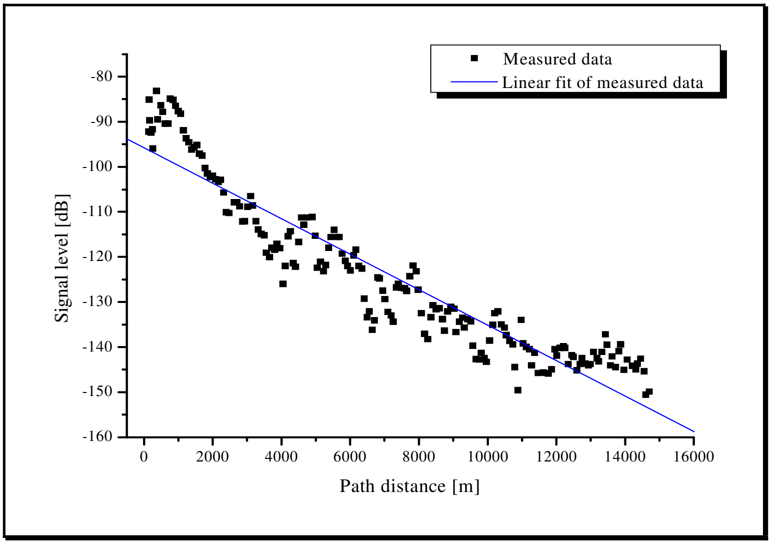


Fig. 2. Path loss slope from measured data

Fig. 3 shows the results calculated using Eq. (2) for the predicted data of the propagation path loss in free space, which is frequently used to predict the propagation path loss over the sea. The propagation path loss and regression analysis according to distance are also presented using the measured data.

Fig. 4 shows the difference of the regression of measured data against predicted data of propagation path loss in free space.

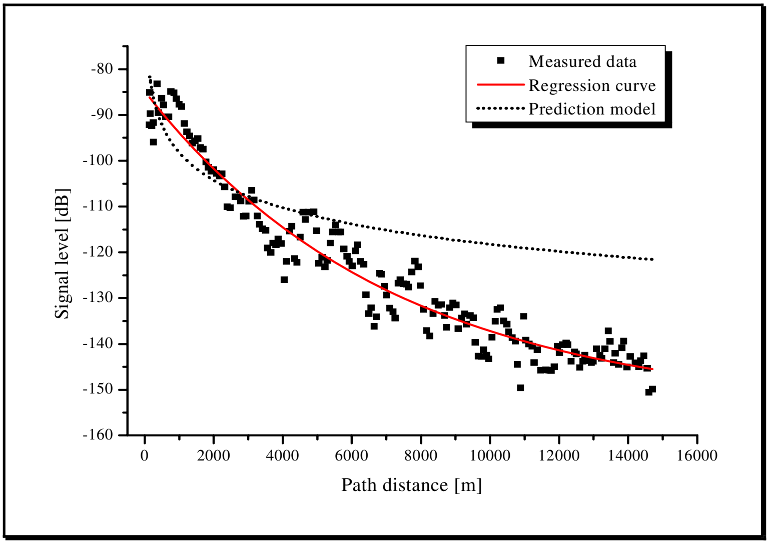


Fig. 3. Propagation path loss according to distance

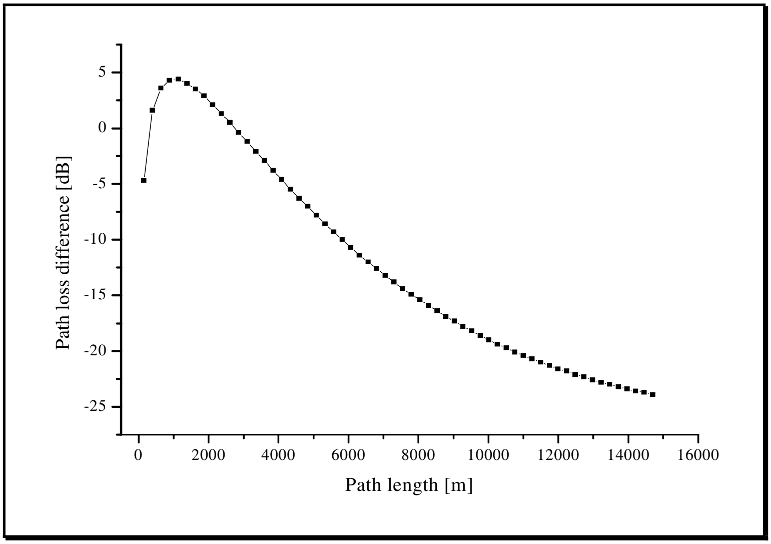


Fig. 4. Path loss difference according to distance

The results in Fig. 2, 3 and 4 suggest that the measured data of the propagation path loss over the sea surface were smaller than the predicted data of the propagation path loss in free space up to 2,200m, but bigger at a distance above 2,200m. As the path length was increased, the measured data were greatly increased compared to the path loss of predicted data. The smaller of the measured propagation path loss up to

2,200m, as compared to the predicted ones, may result from the absence of obstacles in the area, leading to a strong radio strength and large radio filed strength of the reflected radio signal. More specifically, the propagation path loss over the sea surface was about 40dB/dec, which is 20dB/dec bigger than the propagation path loss in free space of 20dB/dec.

Table 3 and 4 show the difference and the standard deviations of the predicted data of the propagation path loss in free space.

From the experimental data, we know that one standard deviation of data spread on any radio path length is about 8dB. This spread is due to the various terrain conditions

Table 3. Difference between predicted and measured data according to distance

Path length [m]	Difference range [dB]	Min. difference [dB]	Max. difference [dB]
0 ~ 2000	-10.7 ~ +11.6	0.7	11.6
2001 ~ 4000	-10.6 ~ +2.3	0.1	10.6
4001 ~ 6000	-15.6 ~ +0.8	0.2	15.6
6001 ~ 8000	-21.5 ~ -4.4	4.4	21.5
8001 ~ 10000	-25.1 ~ -13.9	13.9	25.1
10001 ~ 12000	-30.6 ~ -13.6	13.6	30.6
12001 ~ 14000	-24.9 ~ -16.4	16.4	24.9
14001 ~ 14700	-29.1 ~ -21.2	21.2	29.1

Table 4. Standard deviation of predicted and measured data according to distance

Path length [m]	Measured data & predicted data	Regression data & predicted data
0 ~ 2000	6.3	3.0
2001 ~ 4000	3.9	1.8
4001 ~ 6000	4.5	1.9
6001 ~ 8000	4.8	1.5
8001 ~ 10000	3.9	1.2
10001 ~ 12000	4.6	0.9
12001 ~ 14000	2.0	0.6
14001 ~ 14700	3.0	0.2
Total	**10.3**	**9.3**

from which the data are collected at the same radio path length [7, 8, 9]. However, the standard deviation of the measured data for the propagation path loss over the sea surface is 10.3dB and the regression is 9.3dB, as compared to the predicted data for the propagation path loss in free space.

5 Conclusions

It is common practice to use the predicted model of the propagation path loss in free space to predict the propagation path loss over the sea. Thus, in this study, we measured the propagation path loss of a 1950 MHz band signal over the sea, and compared the results to the predicted data of the propagation path loss in free space. The principal results of this comparison are as follows.

- The propagation path loss over the sea surface was about 40dB/dec, which was 20dB/dec bigger than the propagation path loss in free space (20dB/dec).
- The Standard deviation of the predicted and measured data for the propagation path loss over the sea surface is 10.3dB, which is bigger than the standard deviation of the propagation loss land (8dB).
- As path length was increased, the differences were greatly increased.

The measured results reported in this paper are very valuable in that they provided a means of determining the optimum base-station locations, suitable data rates and estimating their coverage, without having to conduct extensive propagation measurements, which are very expensive and time consuming. Further studies are needed to develop the propagation prediction model for above the sea, by measuring the propagation path loss over the sea surface for various frequency bands.

Acknowledgment. I would like thank Dea-sick Choi, Jin-man Kim, Kyung-Jae Kim (RF Engineering Dept. R&D Center LGE) for useful discussions and valuable data.

References

1. Tapan K. Sarkar, Zhong Ji, Kyungjung Kim, Abdellatif Medouri and Magdalena Salazar-Palma, "A Survey of Various Propagation Models for Mobile Communication," IEEE Antennas and Propagation Magazine, Vol. 45, No. 3 (June 2003) 51-82
2. Ki-sun Kim et al., Mobile Cellular Telecommunication (Analog and Digital Systems-2nd), Sigmapress (1996) 158-164
3. Dea-sick Choi, Jin-man Kim, Kyung-Jae Kim, "An Analysis of Radio Propagation Path Loss in the Sea," KEES proceeding, Vol. 10, No 1 (November, 2000) 255-258
4. Dr. Kamilo Feher, Wireless Digital Communications: Modulation & Spread Spectrum Applications, Prentice Hall Inc., (1995) 66-69
5. Seung-min Wee, Si-hwa Kim and Il-dong Chang, "On the Implementation of Route Planning Algorithms on the Electronic Chart system," Journal of Korea Institute of Navigation, Vol. 24, No. 3 (2000) 167-176

6. Weon-jae Yang, Seung-hwan Jun and Gei-kak Park, “Development of GPS simulation Tool Kit for personal computer,” Journal of Korea Institute of Navigation, Vol. 24, No. 4 (2000) 219-226
7. William C. Y. Lee, Mobile Communications Design Fundamentals, John Wiley & Sons Inc.(1993) 51-53
8. William C. Y. Lee, Mobile Communications Engineering, McGraw Hill Book Co. (1982) 107
9. K. K. Kelly Ⅱ, “Flat Suburban Area Propagation of 820 MHz,” IEEE Transactions on Vehicular Technology, Vol. Vt-27 (November 1978) 198-204

Workload Loss Examinations with a Novel Probabilistic Extension of Network Calculus

András Gulyás and József Bíró

Budapest University of Technology and Economics,
Department of Telecommunications and Media Informatics,
1117 Budapest, Magyar tudósok körútja 2., Hungary
{gulyas, biro}@tmit.bme.hu

Abstract. The estimation of the expected traffic loss ratio (workload loss ratio, WLR) is a key issue in provisioning Quality of Service in packet based communication networks. Despite of its importance, the stationary (long run) loss ratio in queueing analysis is usually estimated through other assessable quantities, typically based on the approximates of the buffer overflow probability. In this paper we define a calculus for communication networks which is suitable for workload loss estimation based on the original definition of stationary loss ratio. Our novel calculus is a probabilistic extension of the deterministic network calculus, and takes an envelope approach to describe arrivals and services for the quantification of resource requirements in the network. We introduce the effective w-arrival curve and the effective w-service curve for describing the inputs and the service and we show that the per-node results can be extended to a network of nodes with the definition of the effective network w-service curve.

Keywords: Network calculus, resource estimation, statistical multiplexing.

1 Introduction

Real time applications in today and future heterogeneous networking environment require simple and efficient Quality of Service provisioning. The expected traffic (packet) loss ratio at network nodes is one of the key QoS parameters which should always be considered and controlled in almost all kind of traffic. Traffic management functions (like connection admission control, packet scheduling algorithms) strongly rely on loss performance analysis.

During the past few years significant attention has been paid for bounding the workload loss ratio within the framework of deterministic network calculus [1]. In [2, 3] some long run loss ratio bounds have been presented, which are founded on buffer saturation probability approximations, hence we call them indirect bounds[1]. More recently in [7] [8] a definition based stochastic workload loss bounding technique has been proposed for deterministic network calculus.

[1] It is true in general, that most of the papers concerning loss ratio apply buffer overflow probability for WLR estimation [4], [5], nevertheless, it is shown, that the ratio $\frac{WLR}{Pr(Q>q)}$ can be arbitrary under certain circumstances [6].

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 533–544, 2006.

Since the worst-case view of the deterministic network calculus results in an overestimation of the actual resource requirements of traffic flows in a packet network, the extension of the network calculus to a probabilistic setting receives a significant attention nowadays [2, 9, 10, 11, 12, 13]. The existing probabilistic extensions share a common property that they assign some kind of violation probability to the definitions of the arrival and service curves. This property makes the estimation of the the overflow type quantities much easier as is shown in [14], however such extensions are not suitable for the direct estimation of the workload loss ratio which still has to be done in an indirect way. These complications indicate, that the workload loss ratio bounds cannot be deduced from the current stochastic versions of network calculus in a straightforward manner [8]. This fact urged us to compose the problem in a more natural way.

Our paper is organized as follows: In section 3 a short overview of deterministic network calculus is given followed by the most important results of a recently introduced min-plus algebra [15] [1] based stochastic extension [10] to the deterministic network calculus. After that, a novel calculus is defined which is designed for direct (definition based) workload loss ratio approximations. We introduce the effective w-arrival curve and the effective w-service curve for describing the inputs and the service and we prove fundamental per-node statements for the backlog, delay and the effective w-arrival curve of the output traffic. It will be shown that the per-node results can be extended to a network of nodes with the definition of the effective network w-service curve in section 4. The connection between the effective w-arrival curve and effective bandwidth [16], is pointed out in section 5. In section 6 we compare the derived workload loss bound with the closest existing probabilistic direct bound [7] and some simulation results.

2 Notation and Assumptions

In this paper the following notations are used: $A_i(s,t]$[2] denotes the number of bits arrived to a a node from flow i and $D_i(s,t]$ the output of flow i from the node within the interval (s,t]. If we use $A_i(t)$ and $D_i(t)$ that will mean $A_i(0,t]$ and $D_i(0,t]$ respectively. If a node has I inputs $A_I(t) := \sum_{i=1}^{I} A_i(t)$, and $D_I(t) := \sum_{i=1}^{I} D_i(t)$. The backlog at time t is given by $B(t) = A(t) - D(t)$ and the delay at time t is given by $W(t) = \inf\{d \geq 0 : A(t-d) \leq D(t)\}$. In a network context we denote by $A^N(t)$ and $D^N(t)$ the arrivals and departures in node N. Subscripts and superscripts are dropped whenever possible to simplify the notation. Let $f \otimes g(t) = \inf_{0 \leq s \leq t}\{f(t-s) + g(s)\}$ denote the min-plus convolution and $f \oslash g(t) = \sup_{0 \leq u}\{f(t+u) - g(u)\}$ the min-plus deconvolution of functions f and g as it is defined in the min-plus algebra [15] [1]. We define the positive part operator as $(expr)^+ = \max[expr, 0]$. For the theorems assume that $A_1, A_2, ..., A_I$ are independent and A_i and D_i are stationary and ergodic.

[2] Without loss of generality we consider a bit-processing system, since it can be shown, that the result can be applied for systems with higher granularity (cells, packets).

3 Theoretical Background

Network calculus is a method to determine resource requirements of traffic flows by taking an envelope approach to describe arrivals and services in the network. One of the first applications of this type of analysis to computer networks was given in [17] and extensions can be found in [15] [1]. In the followings we recall the fundamental results.

3.1 Deterministic Network Calculus

In the deterministic network calculus the characteristics of the input sources are described in terms of arrival curves and the offered service from the nodes are given by the so called service curves. In the followings we recall the exact definitions of these notions from [1]:

Definition 1 (Arrival curve [1]). *We say that a given arrival process $A(t)$ has α as an arrival curve if for all $t > s$:*

$$A(t) - A(s) \leq \alpha(t - s) \tag{1}$$

Definition 2 (Service curve [1]). *Consider a node N and a flow through N with input and output function $A(t)$ and $D(t)$. We say that N offers to the flow a service curve β if and only if*

$$D(t) \geq A \otimes \beta(t). \tag{2}$$

The greatest advantage of the deterministic network calculus is the applicability of the per node results to the concatenation of several nodes. This happens through the definition of the network service curve which express the offered service from a network of nodes. If the hth node within the route ($h = 1, 2, ..., H$) of nodes offers to a flow a service curve β_h, then the network service curve can be expressed as $\beta_{net} = \beta_1 \otimes \beta_2 \otimes ... \otimes \beta_H$.

However the deterministic network calculus is a powerful and expressive tool for describing the properties of communication networks, its worst-case system view cannot take the effects of the statistical multiplexing into consideration. This fact usually leads to the overestimation of the resource requirements of multiplexed traffic sources.

3.2 Probabilistic Extensions of the Deterministic Network Calculus

In order to benefit from the statistical multiplexing several probabilistic extensions of the deterministic network calculus have been elaborated in the past few years. The common property of these studies, that they assign a bound on the violation probability that the incoming traffic exceeds its statistical envelope. For example in [13] we found assumptions that the inputs have stochastically bounded burstiness, in [11] the authors assume that the moment generating functions of the inputs are exponentially bounded. Probabilistic extensions of the network calculus are usually referred as statistical network calculus. Since our novel calculus relies on the min-plus algebra we recall here the results of the only statistical network calculus approach that is based on the min-plus algebra [10]. This calculus defines the effective envelope for the arrival processes.

Definition 3 (Effective envelope [10]). *An effective envelope for an arrival process A is a non-negative function G such that for all t and τ:*

$$P\left\{A(t+\tau)-A(t)\leq G^{\varepsilon}(\tau)\right\}>1-\varepsilon \tag{3}$$

To characterize the available service to a flow or to multiplexed flows the effective service curve is used which can be seen as a probabilistic measure of the available service.

Definition 4 (Effective service curve [10]). *Given an arrival process A, an effective service curve is a non-negative function S^{ε} that satisfies for all $t\geq 0$:*

$$P\left\{D(t)\geq A\otimes S^{\varepsilon}(t)\right\}\geq 1-\varepsilon \tag{4}$$

The following theorems recall the statistical bounds for the delay, the output envelope and the backlog using the terminology of the min-plus algebra on effective envelopes and effective service curves. As we referred earlier, in order to derive such results appropriate time scale limit assumptions are needed, it is assumed, that the node offers a service curve S^{ε_s} which satisfies the additional requirement that there exists a time scale T such that for all $t\geq 0$:

$$P\left\{D(t)\geq \inf_{\tau\leq T}\left\{A(t-\tau)+S^{\varepsilon_s}(\tau)\right\}\right\}\geq 1-\varepsilon_s \tag{5}$$

For all theorems we assume that G^{ε} is an effective envelope for the arrivals A to a node and we have a $T<\infty$ in (5). Define $\varepsilon_\omega := \varepsilon_s + T\varepsilon$.

Theorem 1 (Output traffic envelope [10]). *The function $G^{\varepsilon}\oslash S^{\varepsilon_s}$ is an effective envelope for the output traffic.*

Theorem 2 (Backlog bound [10]). *$G^{\varepsilon}\oslash S^{\varepsilon_s}(0)$ is a probabilistic bound on the backlog, in the sense that, for all $t\geq 0$,*

$$P\left\{B(t)\leq G^{\varepsilon}\oslash S^{\varepsilon_s}(0)\right\}\geq 1-\varepsilon_\omega \tag{6}$$

Theorem 3 (Delay bound [10]). *If $d\geq 0$ satisfies that $\sup_{\tau\leq T}\{G^{\varepsilon}(\tau-d)-S^{\varepsilon_s}(\tau)\}\leq 0$, then d is a probabilistic delay bound in the sense that, for all $t\geq 0$:*

$$P\left\{W(t)\leq d\right\}\geq 1-\varepsilon_\omega \tag{7}$$

Similar to the deterministic calculus the effective service curve of a network can be expressed as the convolution of the service at each node. Consider a network of nodes where the hth node offers an effective service curve $S_h^{\varepsilon_s}$ to a flow. It is assumed that:

$$P\left\{D^h(t)\geq \inf_{\tau\leq T_h}\left\{A^h(t-\tau)+S_h^{\varepsilon_s}(\tau)\right\}\right\}\geq 1-\varepsilon_s \tag{8}$$

Theorem 4 (Effective network service curve [10]). *If the service offered at each node $h = 1, ..., H$ on the path of a flow is given by a service curve $S_h^{\varepsilon_s}$, then an effective network service curve $S_{net}^{\varepsilon_\omega}$ for the flow is given by $S_{net}^{\varepsilon_\omega} = S_1^{\varepsilon_s} \otimes S_2^{\varepsilon_s} \otimes ... \otimes S_H^{\varepsilon_s}$ with a violation probability bounded above by $\varepsilon_\omega = \varepsilon_s \sum_{h=1}^{H} \left(1 + (h-1)T^h\right)$.*

We can see, that these statements for backlog delay etc. are expressed with a straightforward calculation from the defined effective envelopes and service curves, however quantifying packet loss with the existing probabilistic extensions of network calculus is a highly non-trivial problem even in an indirect way [2] [7] [8]. One can also observe, that these statements above rely on an accurate busy period analysis for estimating the appropriate time scale and require that the infimum in (5) and (8) is taken within a finite interval. In the next section we define a statistical network calculus, which is designed for direct packet loss calculations and which application does not require such assumptions for the time scale.

4 A Novel Statistical Network Calculus for Workload Loss Estimations

We can see in (3) and (4) that the definition of the effective envelope and the effective service curve happens by assigning some violation probability to the deterministic arrival and service curves (1) (2). As it was pointed out earlier this approach is favourable for overflow type quantities like buffer overflow probability however quantifying packet loss in a direct way turns out to be non-trivial.

In the followings a novel calculus is defined which is suitable for loss examinations. We set out from the definition of the workload loss ratio which looks like this for stationary and ergodic systems:

$$WLR = \frac{E[\text{\# of lost bits in a unit time interval}]}{E[\text{\# of bits arriving in a unit time interval}]} \le \frac{E[(B-q)^+]}{E[A]} \,. \tag{9}$$

where B represents the stationary backlog of the system with infinite buffer, q is the buffer threshold and $E[A] = E[A(0,1)]$ is the number of bits arriving in a unit time interval[3]. Based on (9) we assign Z^φ and S^{φ_s} functions to the input and the service respectively and we call them effective w-arrival curve and effective w-service curve hereafter.

Definition 5 (Effective w-arrival curve). *We call Z^φ the effective w-arrival curve of the flow with arrival process A if for all t and τ:*

$$E[(A(t+\tau) - A(t) - Z^\varphi(\tau))^+] \le \varphi \tag{10}$$

[3] It is proven (e.g. in [6] and [18]) that the expected value of the number of lost bits in a finite buffer system, can be bounded from above by the number of packets overflown in the system with infinite buffer.

Definition 6 (Effective w-service curve). *For an input with arrival process A a node offers an effective w-service curve S^{φ_s} if for all $t \geq 0$:*

$$E[(A \otimes S^{\varphi_s}(t) - D(t))^+] \leq \varphi_s \tag{11}$$

We note that by letting φ and φ_s to zero the arrival and service curves of the deterministic network calculus can be recovered.

Within the framework of the following theorems we formalize stochastic bounds on some fundamental system characteristics like backlog, delay and output traffic envelope, with min-plus calculus operations on effective w-arrival curves and effective w-service curves. For the proofs the following lemma is needed about the positive part operator:

Lemma 1. *For given X_1, X_2, X_3, X_4 random variables:*

$$E[(X_1 - X_2 + X_3 - X_4)^+] \leq E[(X_1 - X_2)^+] + E[(X_3 - X_4)^+] \tag{12}$$

The proof of this lemma is left to the reader.

Theorem 5 (Statement for the backlog). *$Z^{\varphi} \oslash S^{\varphi_s}(0)$ is a probabilistic bound on the backlog, in the sense that, for all $t \geq 0$,*

$$E[(B(t) - Z^{\varphi} \oslash S^{\varphi_s}(0))^+] \leq \varphi + \varphi_s \tag{13}$$

Proof. It follows from the definition of the backlog that
$E[(B(t) - Z^{\varphi} \oslash S^{\varphi_s}(0))^+] = E[(A(t) - D(t) - Z^{\varphi} \oslash S^{\varphi_s}(0))^+] = E[(A(t) + A \otimes S^{\varphi_s}(t) - D(t) - A \otimes S^{\varphi_s}(t) - Z^{\varphi} \oslash S^{\varphi_s}(0))^+]$.
For any choice of an arbitrarily small $\delta > 0$, there exists a finite s^* such that $A(t - s^*) + S^{\varphi_s}(s^*) < A \otimes S^{\varphi_s}(t) + \delta$ and the whole expression is increased by the substitution of this s^* into the min-plus deconvolution in $Z^{\varphi} \oslash S^{\varphi_s}$, so we get that:
$E[(A(t) + A \otimes S^{\varphi_s}(t) - D(t) - A \otimes S^{\varphi_s}(t) - Z^{\varphi} \oslash S^{\varphi_s}(0))^+] \leq E[(A(t) + A \otimes S^{\varphi_s}(t) - D(t) - A(t - s^*) - S^{\varphi_s}(s^*) + \delta - Z^{\varphi}(s^*) + S^{\varphi_s}(s^*))^+]$.
After simplification we obtain that:
$E[(A(t) - A(t - s^*) + \delta - Z^{\varphi}(s^*) + A \otimes S^{\varphi_s}(t) - D(t))^+]$.
By using Lemma 1 twice we get:
$E[(A(t) - A(t - s^*) + \delta - Z^{\varphi}(s^*) + A \otimes S^{\varphi_s}(t) - D(t))^+] \leq E[(A(t) - A(t - s^*) - Z^{\varphi}(s^*))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+] + E[\delta^+]$.
From the definition of the effective w-arrival curve and the effective w-service curve we recover that:
$E[(A(t) - A(t - s^*) - Z^{\varphi}(s^*))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+] + E[\delta^+] \leq \varphi + \varphi_s + \delta$.
Since δ can be arbitrarily small, letting it shrink to 0 recovers the desired result, which completes the proof. Q.E.D.

The alert reader may notice that the left hand side of (13) express the expected value of the number of bits above a certain buffer level $Z^{\varphi} \oslash S^{\varphi_s}(0)$ in an infinite buffer system. In other words if we imagine a buffered system with a buffer size $Z^{\varphi} \oslash S^{\varphi_s}(0)$ the statement in (13) establishes an upper bound on the loss rate. Dividing this upper bound of the loss rate with the expected value of the bits arriving to the node gives an upper bound on the workload loss ratio.

Theorem 6 (W-arrival curve for the output). *The function $Z^{\varphi} \oslash S^{\varphi_s}$ is an effective w-arrival curve for the output traffic from the node in the sense that for all t and τ:*

$$E[(D(t+\tau) - D(t) - Z^{\varphi} \oslash S^{\varphi_s}(\tau))^+] \leq \varphi + \varphi_s \tag{14}$$

Proof. $E[(D(t+\tau) - D(t) - Z^{\varphi} \oslash S^{\varphi_s}(\tau))^+] = E[(D(t+\tau) + A \otimes S^{\varphi_s}(t) - D(t) - A \otimes S^{\varphi_s}(t) - Z^{\varphi} \oslash S^{\varphi_s}(\tau))^+]$.
Using Lemma 1 and the fact that $A(t+\tau) \geq D(t+\tau)$ we obtain that:
$E[(D(t+\tau) + A \otimes S^{\varphi_s}(t) - D(t) - A \otimes S^{\varphi_s}(t) - Z^{\varphi} \oslash S^{\varphi_s}(\tau))^+] \leq E[(A(t+\tau) - A \otimes S^{\varphi_s}(t) - Z^{\varphi} \oslash S^{\varphi_s}(\tau))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+]$.
For any choice of an arbitrarily small $\delta > 0$, there exists a finite s^* such that $A(t-s^*) + S^{\varphi_s}(s^*) < A \otimes S^{\varphi_s}(t) + \delta$ and the whole expression is increased by the substitution of this s^* into the min-plus deconvolution in $Z^{\varphi} \oslash S^{\varphi_s}$, so we obtain:
$E[(A(t+\tau) - A \otimes S^{\varphi_s}(t) - Z^{\varphi} \oslash S^{\varphi_s}(\tau))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+] \leq E[(A(t+\tau) - A(t-s^*) - S^{\varphi_s}(s^*) + \delta - Z^{\varphi}(\tau+s^*) + S^{\varphi_s}(s^*))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+]$.
After some simplification and applying Lemma 1 we get:
$E[(A(t+\tau) - A(t-s^*) - Z^{\varphi}(\tau+s^*))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+] + E[\delta^+] \leq \varphi + \varphi_s + \delta$. The last step follows from the definition of the effective w-arrival curve and the effective w-service curve. Since δ can be arbitrarily small, letting it shrink to 0 recovers the desired result, which completes the proof. Q.E.D.

Theorem 7 (Statement for the delay). *If $d : Z^{\varphi}(\tau - d) \leq S^{\varphi_s}(\tau)$ for all τ then:*

$$E[A(t-d) - D(t)] \leq \varphi + \varphi_s \tag{15}$$

Proof. $E[A(t-d) - D(t)] \leq E[A(t-d) - A \otimes S^{\varphi_s}(t) + A \otimes S^{\varphi_s}(t) - D(t)] \leq E[(A(t-d) - A \otimes S^{\varphi_s}(t) + A \otimes S^{\varphi_s}(t) - D(t))^+]$.
For any choice of an arbitrarily small $\delta > 0$, there exists a finite s^* such that $A(t-s^*) + S^{\varphi_s}(s^*) < A \otimes S^{\varphi_s}(t) + \delta$ and the whole expression is increased by the substitution of this s^* into the first min-plus convolution:
$E[(A(t-d) - A \otimes S^{\varphi_s}(t) + A \otimes S^{\varphi_s}(t) - D(t))^+] \leq E[(A(t-d) - A(t-s^*) - S^{\varphi_s}(s^*) + \delta + A \otimes S^{\varphi_s}(t) - D(t))^+]$
From Lemma 1 it follows that:
$E[(A(t-d) - A(t-s^*) - S^{\varphi_s}(s^*) + \delta + A \otimes S^{\varphi_s}(t) - D(t))^+] \leq E[(A(t-d) - A(t-s^*) - S^{\varphi_s}(s^*))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+] + E[\delta^+]$.
It follows from the additional assumption of the theorem that:
$E[(A(t-d) - A(t-s^*) - S^{\varphi_s}(s^*))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+] + E[\delta^+] \leq E[(A(t-d) - A(t-s^*) - Z^{\varphi}(s^*-d))^+] + E[(A \otimes S^{\varphi_s}(t) - D(t))^+] + E[\delta^+] \leq \varphi + \varphi_s + \delta$
The last step follows from the definition of the effective w-arrival curve and the effective w-service curve. Since δ can be arbitrarily small, letting it shrink to 0 recovers the desired result, which completes the proof. Q.E.D.

One can notice that Theorem 7 establishes a bound on the expected value of the number of bits that suffers from a delay larger than d. In order to establish end-to-end bounds from the single node results we are going to express the effective w-service curve of a network of nodes. In the following theorem the effective w-service curve of two concatenated nodes is given. Let $S_N^{\varphi_i}$ mean the effective w-arrival curve of input process A_i at node N.

Theorem 8 (Concatenation of nodes). *Assume that a flow traverses nodes N_1 and N_2 in sequence. If $E[(A^{N1} \otimes S_{N1}^{\varphi_1}(t) - A^{N2}(t))^+] \leq \varphi_1$ and $E[(A^{N2} \otimes S_{N2}^{\varphi_2}(t) - D^{N2}(t))^+] \leq \varphi_2$, then*

$$E[(A^{N1} \otimes S_{N1}^{\varphi_1} \otimes S_{N2}^{\varphi_2}(t) - D^{N2}(t))^+] \leq \varphi_1 + \varphi_2 \tag{16}$$

which means, that $S_{N1}^{\varphi_1} \otimes S_{N2}^{\varphi_2}$ is a stochastic w-service curve for the system which consists of the concatenation of these two nodes with $\varphi_1 + \varphi_2$ parameter.

Proof. $E[(A^{N1} \otimes S_{N1}^{\varphi_1} \otimes S_{N2}^{\varphi_2}(t) - D^{N2}(t))^+] = E[(A^{N1} \otimes S_{N1}^{\varphi_1} \otimes S_{N2}^{\varphi_2}(t) - A^{N2} \otimes S_{N2}^{\varphi_2}(t) + A^{N2} \otimes S_{N2}^{\varphi_2}(t) - D^{N2}(t))^+]$.
From Lemma 1 it follows that:
$E[(A^{N1} \otimes S_{N1}^{\varphi_1} \otimes S_{N2}^{\varphi_2}(t) - A^{N2} \otimes S_{N2}^{\varphi_2}(t) + A^{N2} \otimes S_{N2}^{\varphi_2}(t) - D^{N2}(t))^+] \leq E[(A^{N1} \otimes S_{N1}^{\varphi_1} \otimes S_{N2}^{\varphi_2}(t) - A^{N2} \otimes S_{N2}^{\varphi_2}(t))^+] + E[(A^{N2} \otimes S_{N2}^{\varphi_2}(t) - D^{N2}(t))^+]$.
Using the definition on the min-plus convolution and the effective w-service curve we recover that :
$E[(A^{N1} \otimes S_{N1}^{\varphi_1} \otimes S_{N2}^{\varphi_2}(t) - A^{N2} \otimes S_{N2}^{\varphi_2}(t))^+] + E[(A^{N2} \otimes S_{N2}^{\varphi_2}(t) - D^{N2}(t))^+] \leq E[(\inf_{0\leq s\leq t}\{\inf_{0\leq u\leq t-s}\{A^{N1}(t-s-u) + S_{N1}^{\varphi_1}(u)\} + S_{N2}^{\varphi_2}(s)\} - \inf_{0\leq s\leq t}\{A^{N2}(t-s) + S_{N2}^{\varphi_2}(s)\})^+] + \varphi_2$.

For any choice of an arbitrarily small $\delta > 0$, there exists a finite s^* such that $A^{N2}(t-s^*) + S_{N2}^{\varphi_2}(s^*) < \inf_{0\leq s\leq t}\{A^{N2}(t-s) + S_{N2}^{\varphi_2}(s)\} + \delta$ we get:
$E[(\inf_{0\leq s\leq t}\{\inf_{0\leq u\leq t-s}\{A^{N1}(t-s-u) + S_{N1}^{\varphi_1}(u)\} + S_{N2}^{\varphi_2}(s)\} - \inf_{0\leq s\leq t}\{A^{N2}(t-s) + S_{N2}^{\varphi_2}(s)\})^+] + \varphi_2 \leq E[(\inf_{0\leq u\leq t-s^*}\{A^{N1}(t-s^*-u) + S_{N1}^{\varphi_1}(u)\} + S_{N2}^{\varphi_2}(s^*) - A^{N2}(t-s^*) - S_{N2}^{\varphi_2}(s^*) + \delta)^+] + \varphi_2 = E[(A^{N1} \otimes S_{N1}^{\varphi_1}(t-s^*) - A^{N2}(t-s^*) + \delta)^+] + \varphi_2$.
Applying Lemma 1 and using the definition of the effective w-service curve we get:
$E[(A^{N1} \otimes S_{N1}^{\varphi_1}(t-s^*) - A^{N2}(t-s^*) + \delta)^+] + \varphi_2 \leq \varphi_1 + \varphi_2 + \delta$.
Since δ can be arbitrarily small, letting it shrink to 0 recovers the desired result, which completes the proof. Q.E.D.

The application of Theorem 8 iteratively to a network of nodes the gives the following corollary.

Corollary 1 (Effective network w-service curve). *If the service offered at each node $h = 1, ..., H$ on the path of a flow is given by an effective w-service curve $S_h^{\varphi_{sh}}$, then an effective network w-service curve $S_{net}^{\varphi_\omega}$ for the flow is given by:*

$$S_{net}^{\varphi_\omega} = S_1^{\varphi_{s1}} \otimes S_2^{\varphi_{s2}} \otimes ... \otimes S_H^{\varphi_{sH}} \tag{17}$$

with a parameter:

$$\varphi_\omega = \sum_{h=1}^{H} \varphi_{sh} \tag{18}$$

Using corollary 1 we are able to draw up end-to-end workload loss ratio bounds according to Theorem 13.

5 The Effective w-Arrival Curve and the Effective Bandwidth

The theory of effective bandwidth [16] defines a framework for service provisioning, that describes the minimum bandwidth requirement of a traffic source in terms of the effective bandwidth, which is a probabilistic quantity between the average and peak rate of the input source. This concept provides a measure of resource usage which takes proper account of the varying statistical characteristics and QoS requirements of traffic sources. A widely referenced definition of effective bandwidth is the following.

Definition 7 (Effective bandwidth [16]). *The effective bandwidth of the source with arrival process $A(t)$ is defined as:*

$$\alpha_e(s,\tau) = \sup_{t\geq 0}\left\{\frac{1}{st}\log E[e^{s(A(t+\tau)-A(t))}]\right\}, 0 < s, \tau < \infty. \tag{19}$$

The following theorem makes contact between the effective w-arrival curve and the effective bandwidth.

Theorem 9

$$Z^{\varphi}(\tau) = \inf_{s>0}\{\tau\alpha_e(s,\tau) - \frac{\log(\varphi s)}{s}\} \tag{20}$$

Proof.

$$E[(A(t+\tau) - A(t) - Z^{\varphi}(\tau))^+] \leq \frac{e^{s(-Z^{\varphi}(\tau)+\tau\alpha_e(s,\tau))}}{s} \tag{21}$$

for all values of s. Let φ defined as:

$$\frac{e^{s(-Z^{\varphi}(\tau)+\tau\alpha_e(s,\tau))}}{s} := \varphi. \tag{22}$$

For $Z^{\varphi}(\tau)$ we obtain:

$$Z^{\varphi}(\tau) = \tau\alpha_e(s,\tau) - \frac{\log(\varphi s)}{s}. \tag{23}$$

By taking the infimum over s we obtain the smallest effective w-arrival curve:

$$Z^{\varphi}(\tau) = \inf_{s>0}\left\{\tau\alpha_e(s,\tau) - \frac{\log(\varphi s)}{s}\right\}. \tag{24}$$

Since the effective bandwidth expressions of various traffic sources have been developed in the last decade the effective w-arrival curve for those sources can be calculated according to Theorem 9. For demonstration the effective w-arrival curve of multiplexed regulated input flows is shown in Figure 1. The w-arrival curve is normalized by the number of flows and the per flow deterministic arrival curve is also shown for easier interpretation of the figure. One can see that the effective w-arrival curve exploits a significant statistical multiplexing gain.

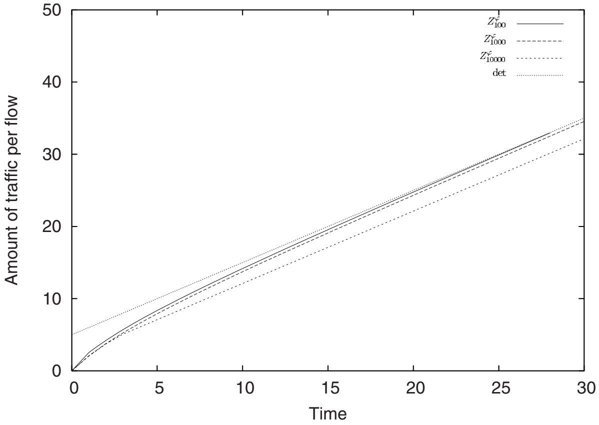


Fig. 1. The statistical multiplexing gain

6 Numerical Results

In this section we investigate the novel workload loss ratio bound deduced from our novel statistical calculus and compare it with the best existing deterministic calculus based probabilistic bound [7] and also with simulation results under NS2. For analysis the following scenario is used. We have 100 input flows, which are token bucket constrained with some deterministic arrival curve $\alpha_i(t) = \bar{\alpha}_i t + \sigma_i$ ($\bar{\alpha}_{1..50} = 133.3$, $\sigma_{1..50} = 8$, $\bar{\alpha}_{51..100} = 66.6$, $\sigma_{51..100} = 5$), and the packet forwarder satisfies a rate latency service curve property, with $\beta(t) = 12500 \cdot \max(t - 8 \cdot 10^{-5}, 0)$, in a work-conserving manner[4]. The sustainable rate of the inputs and the size of the bucket is given in packets and the service rate is given in packets during a second (pps). These parameter values are close to many practical, common applications.

Based on the effective bandwidth for regulated inputs in [16] we use the following formula for the calculation of the effective w-arrival curves in accordance with equation (24):

$$Z^{\varphi}(t) = \inf_{s>0} \left\{ \sum_{i \in \mathcal{I}} \frac{1}{s} \log\left(1 + \frac{\bar{\alpha}_i t}{\alpha_i(t)} \left(e^{(s\alpha_i(t))} - 1 \right) \right) - \frac{\log(\varphi s)}{s} \right\}. \tag{25}$$

The calculation of the workload loss ratio happens according to Theorem 5.

For simulation purposes we made an implementation of the evaluation scenario under the NS2 network simulator [19]. We used random packet generators as inputs, which send packet to the server through a token bucket traffic regulator. For the token bucket regulator we used the Differentiated Services module of the NS2 and set the bucket size and the token generating rate according to

[4] For the proper comparison of the performance of the arrival and w-arrival curves the same deterministic service curve is used for the server.

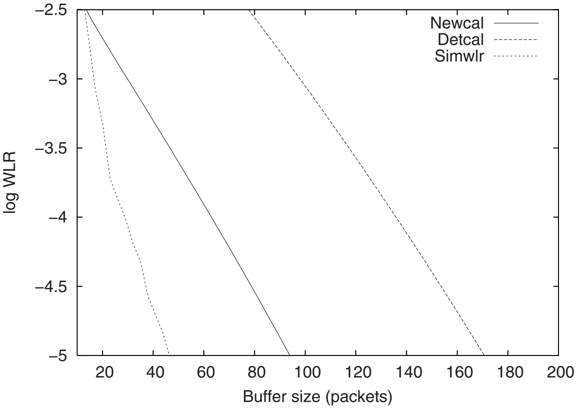


Fig. 2. The comparison of the bounds and the simulation results

the values of the input scenario. The server was a non-preemptive constant rate server with the appropriate service rate. Besides the 100 inputs we set up another packet generator, which sends lower priority packets to the server with the same packet size. This way we ensured the given rate-latency service curve for the input flows among realistic conditions, since there is no service for the higher priority packets, while the server finishes the inchoate. The interesting case from the point of the packet loss is when the inputs exploit the entire input profile, so we set up the packet generators to generate different traffic bursts of alternating sizes with exponentially distributed random inter arrival times. We also controlled the average rate of the generators in order to meet the maximum input rate requirement. We run the simulation ten times for some queue sizes and took the average of the results. Figure 2 show the results of the bounds and the simulation.

We can observe that the novel bound provides a significant improvement of the existing closest result. Comparing with the simulation we state that within the range of interest $(10^{-3} - 10^{-6})$ the result of Theorem 5 gives a considerably well bound on the workload loss ratio.

7 Conclusions

In the focus of this paper was to establish a novel probabilistic calculus for packet networks which is designed for direct workload loss ratio approximations. We introduced the effective w-arrival curve and the effective w-service curve and proved fundamental statements about the backlog, delay and output traffic envelope. We also showed that the per-node results can be carried over to a network of nodes with the definition of the effective network w-service curve and a performance evaluation was given on the workload loss ratio bound that follows from our new theory. Besides these fundamental results our novel calculus raises a lot of questions that have to be answered. The determination of the effective w-service curve for various packet schedulers is a possible topic of further research.

References

1. Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus: A theory of deterministic queuing systems for the Internet*. Springer, 2002.
2. Milan Vojnovic and Jean-Yves Le Boudec. Stochastic analysis of some expedited forwarding networks. *IEEE INFOCOM New York*, June 2002.
3. M. Vojnovic and J. Y. Le Boudec. Stochastic analysis of some expedited forwarding networks. Technical Report DSC/2001/039, EPFL-DI-ICA, July 2001.
4. N. G. Duffield, J. T. Lewis, N. O'Connel, R. Russel, and F. Foomey. Entropy of atm traffic streams: tool for estimating qos parameters. *IEEE Journal of Selected Areas in Communications vol. 13*, March 1995.
5. M. Krunz and A. M. Ramasamy. The correlation structure for a class of scene-based video models and its impact on the dimensioning of video buffers. *IEEE Trans. Multimedia vol. 2*, July 2000.
6. A. György and T. Borsos. Estimates on the packet loss ratio via queue tail probabilities. *IEEE Globecom*, March 2001.
7. András Gulyás, J. Bíró, and Z. Heszberger. A novel direct upper approximation for workload loss ratio in general buffered systems. In *IFIP Networking 2005*, page 718, Waterloo, Canada, May 2005.
8. András Gulyás and J. Bíró. Direct and indirect methods for packet loss estimation in buffered systems. In *EuroNGI 2005*, Rome, Italy, April 2005.
9. A. Burchard R. R. Boorstyn, J. Liebeherr, and C. Oottamakorn. Statistical service assurances for traffic scheduling algorithms. *IEEE Journal on Selected Areas in Communications*, December 2000.
10. A. Burchard, J. Liebeherr, and S. D. Patek. A calculus for end-to-end statistical service guarantees. Technical Report CS-2001-19, University of Virginia, May 2002.
11. C. S. Chang. On deterministic traffic regulation and service guarantees: A systematic approach by filtering. *IEEE Transactions on Information Theory, Vol. 44, pp. 1097-1110*, May 1998.
12. S. Ayyorgun and R. Cruz. A service curve model with loss. Technical Report LA-UR-03-3939, Los Alamos National Laboratory, June 2003.
13. D. Starobinski and M. Sidi. Stochastically bounded burstiness for communication networks. *IEEE Transactions on Information Theory*, January 2000.
14. Chengzhi Li, A. Burchard, and J. Liebeherr. A network calculus with effective bandwidth. Technical Report CS-2003-20, University of Virginia, November 2003.
15. C. S. Chang. Stability, queue length and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, May 1994.
16. F. P. Kelly. Notes on effective bandwidth. *Stochastic Networks: Theory and Applications vol. 4*, Sep 1995.
17. Rene Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications, 13(6):1048-1056*, Aug 1995.
18. H. Kim and N. B. Shroff. Loss probability calculations and asymptotic analysis for finite buffer multiplexers. *IEEE/ACM Trans. on Networking*, 9(6):755–768, Dec 2001.
19. Network Simulator v2. http://www.isi.edu/nsnam/ns/.

Optimized Handoff Decision Mechanisms for Scalable Network Mobility Support*

Sangwook Kang, Yunkuk Kim, Woojin Park, Jaejoon Jo, and Sunshin An

Dept. of Electronics & computer Eng., Korea University,
1, 5-Ga, Anam-dong Sungbuk-ku, Seoul, Korea, Post Code: 136-701
{Klogic, dbs1225, progress, jjj, sunshin}@dsys.korea.ac.kr

Abstract. Network Mobility (NEMO) is concerned with managing the mobility of an entire network and included one or more Mobile Routers (MRs) which are connected as gateways to the Internet. This paper proposes the optimal handoff decision mechanisms, not only avoiding the 'ping-pong effect' and frequent handoffs, but also reducing handoff latency under multi-hop network mobility. The simulation results demonstrate that the proposed method is well adapted for supporting network mobility over traditional handoff decision algorithms.

1 Introduction

Compared to approach like Mobile IPv6 (MIPv6) [1] where each host has the mobility support, NEtwork MObility (NEMO) [2, 3, 4] is concerned with managing the mobility of an entire network, with a varying point of attachment to the Internet. This type of network topology is referred to as a mobile network (NEMO) and includes one or more Mobile Routers (MRs) which are connected as gateways to the Internet. The typical examples of a mobile network are PANs (Personal Area Networks), networks of sensors deployed in vehicles, and access networks deployed in public transportation to provide Internet access to devices carried by their passengers. The Internet Engineering Task Force (IETF) NEMO W/G [5] is developing a solution based on MIPv6 with minimal extensions for this mobile network.

In this paper we consider the situation where several mobile networks are deployed in close vicinity and MRs also tend to change their direction of movement very rapidly (e.g. vehicles). In [5], when a MR leaves a network and enters another network, it should perform the handoff operations like MIPv6. Generally a handoff can be decided by the movement detection algorithms [6, 7, 8], such as Lazy Cell Switching (LCS) and Eager Cell Switching (ECS). The first algorithm, LCS, is based upon the lifetime of the advertisement sent by the router. If a Mobile Node (MN) fails to receive another advertisement from its current network within the specified lifetime, MN should assume it has moved out of range from that network. On the other hand, a handoff in ECS is initiated as soon as a new network is discovered. That is, when a

* 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).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 545 – 555, 2006.

MN detects an advertisement with a different network identifier than the current network, MN assumes that a handoff has happened. These approaches may seem simple and effective to decide a handoff. However, these approaches are unsuitable in NEMO environments where a MR is moving rapidly with a random direction and many MRs are deployed in close vicinity. The first approach can result in a problem called the "ping-pong effect", and the second method may result in considerable handoff latency. Thus, a new handoff decision algorithm is necessary to support optimal handoff operation in multi-NEMO environments.

Furthermore, if a MR which has a wireless transceiver is located more than one hop away from the Access Router (AR), i.e., outside of the propagation scope of AR, MR cannot access directly to the AR and should use its neighbor MRs to access the Internet. But, when MR is located within range of two or more neighbors (ARs or MRs) and is receiving advertisements from all of them, MR cannot distinguish which router is its parent, i.e., the upstream router that destined to the Internet [9]. That is to say, when a new advertisement is received from neighbors, MR cannot decide whether sender is a fixed AR that directly connected to the Internet or a MR which attached to the AR or an isolated MR. So when a moving MR receives the RA from an isolated MR earlier than the other MR which connected to the Internet, a moving MR may think it is under a fixed network and forward its outgoing packets to that MR. In this situation if a moving MR wants to send the packets to the outside of mobile network, the packets cannot be routed correctly. Accordingly, in order to provide uninterrupted services and continuous communication in such NEMO environments, we must consider mechanisms for efficient handoff support.

To solve these problems described in above, this paper proposes the mechanisms for optimal handoff decision under multi-hop NEMO environments, not only supporting Internet connectivity and reducing the routing overhead, but also avoiding the 'ping-pong effect' and frequent handoffs.

The remainder of this paper is organized as follows. In next section, we describe an extended MIPv6's RA message to discover its default router which has the Internet connectivity and then proposes an optimal NEMO handoff procedure. We also propose a new handoff decision algorithm for NEMO. Section 3 evaluates the proposed NEMO handoff decision algorithms. Finally, section 4 concludes this paper.

2 Optimized Handoff Decision Mechanisms

2.1 Extended Router Advertisement (RA) Message

This section describes an extended MIPv6's RA message to discover its default router which has the Internet connectivity and support network mobility under multi-hop NEMO environments. Extending the Prefix Information option of the RA message is suggested as follows:

First, an extra flag 'H' taken from the 'reserved 1' field is used to distinguish the type of sending router when a MR receives advertisements from neighbors. If this flag is unset, it means that a sender is not a fixed AR but a MR operating away from home. If a normal RA message, as defined in [10], is received, the sender is a fixed

AR connected to the Internet. Accordingly, a moving MR should advertise an extended RA with the ‘H’ flag periodically to neighbors until it returns back to its home network.

Second, a new ‘D’ flag is used to decide whether a sending MR has information regarding connectivity to the Internet. If an advertisement from the new neighbor is received and the ‘D’ flag containing in this message is unset, it indicates that the sending MR does not currently have information about its default router, i.e., the sender is an isolated MR. If this flag is set, it means that sending router has information about the Internet connectivity and the ‘Network Prefix’ field is included a delegate Care-of-Address (DCoA) to access the Internet. In here we use the AR’s address as the DCoA for all MNNs within the NEMOs. In this case the network prefix of the received RA message is computed by the leftmost “Prefix Length” bits of sender’s address.

Finally, the ‘Network Level (NLevel)’ field taken from the ‘reserved 2’ field is defined as the number of hop between a MR and AR, which is used to establish parent-child relationships between MRs. This field is initialized to one by AR’s child-MR, i.e. MR that attached to an AR directly, and its value is increased using the distance from AR.

2.2 Optimal NEMO Handoff Procedure

In this section, the procedure to decide an optimal handoff under multi-hop NEMO environments is described. As illustrated in Fig. 1, an optimal handoff decision can be partitioned into four phases as follows:

Phase I : Check the reachability of the default router

In this phase, MR checks the reachability of the current default router and counts the number of RA missed of default router. MR relies on RA message to know whether it is still attached to its default router. For reachability confirmation, MRs keep a counter that counts the number of RAs missed for its default router.

MRs can assume they have missed at least one advertisement if the RA interval passes without receiving an advertisement from its default router, so MRs increase RA miss counter. If the consecutive missing RAs reach three times, the MR decides that it loses reachability with its default router. In this case a MR must attempt registration with a known neighbor or solicit for the discovery of other neighbors. The number of RAs missed (represented as ‘N’) is used to calculate the Internet Connectivity Strength (ICS) of its default router in Phase III.

Phase II : Neighbor router’s type decision

As explained earlier, a moving MR advertises an extended RA periodically to neighbors until it returns back to its home network. If MR receives a normal RA message, as defined in [10], the sender is a fixed AR connected to the Internet. On the other hand, if an extended RA is received and “D” flag contained in this message is set, the sender which sent this message is a MR which had already attached to the Internet. If the ‘D’ flag contained in an extended RA is unset, the sender is an isolated MR.

Once the decision of sender’s type is completed, the process of ICS calculation for handoff decision is initiated as illustrated in Fig. 1.

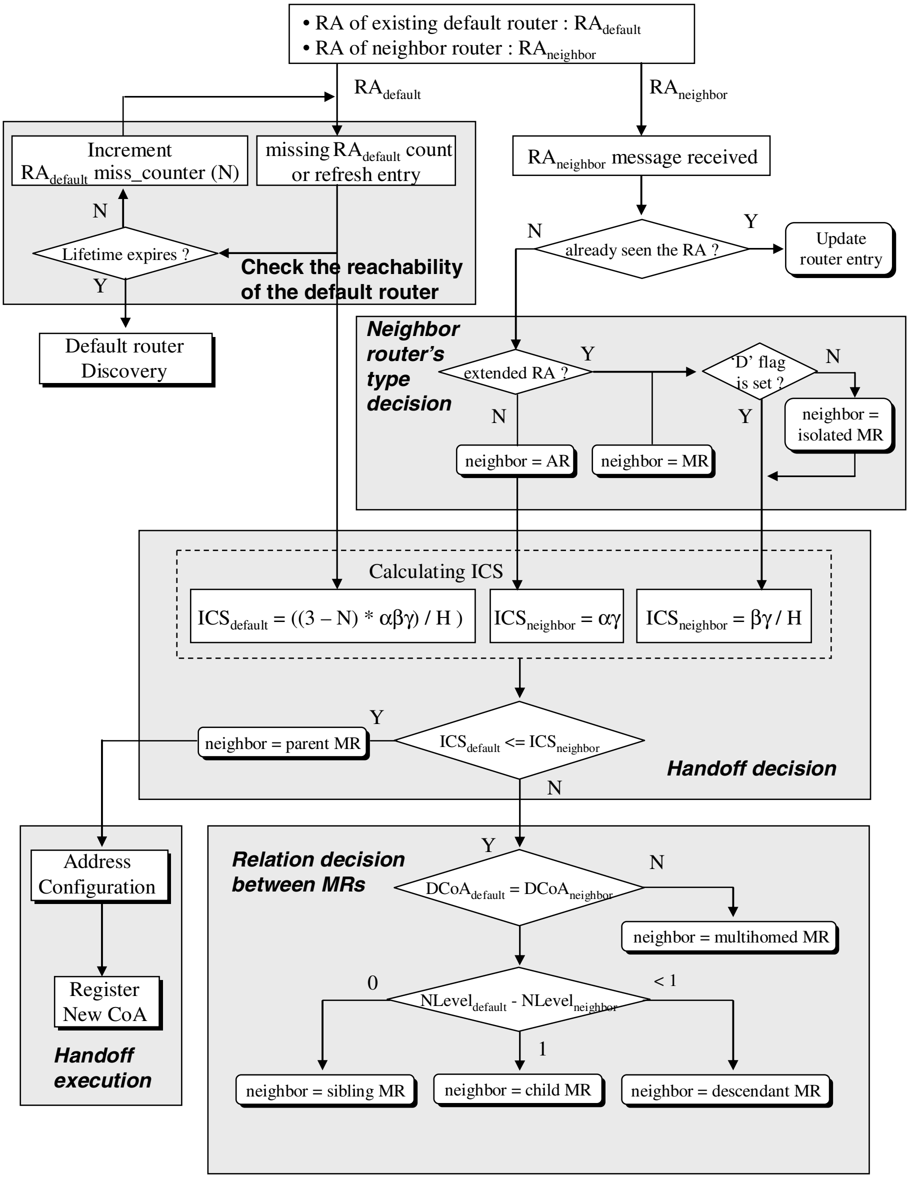


Fig. 1. Decision tree for optimized NEMO handoff

Phase III : Handoff decision and execution

In this phase, MR calculates the ICS of its default and neighbor router, respectively, and then determines whether it performs handoff decision process based on the values of ICS calculated in previous phase. If the ICS of the neighbor is greater than that of its default router, a handoff will be performed. If handoff execution is determined, MR performs handoff process, such as address configuration and register new CoA. Otherwise, MR will proceed with the phase of relation decision between MRs. The handoff decision algorithm for NEMO is discussed in detail in the section 2.3.

Phase IV : Relation decision between MRs

This phase describes the process of relation establishment between MRs. In our proposal, each MR can know both the DCoA, i.e., AR's address to access the Internet, and its network level thanks to information included in an extended RA message. Accordingly, every time the MR receives the RAs sent by new MRs in its coverage area, it performs the operation of relation establishment between MRs as following:

1) If $\text{DCoA}_{\text{neighbor}} = \text{DCoA}_{\text{default}}$ and $\text{NLevel}_{\text{neighbor}} = \text{NLevel}_{\text{default}} + 1$, a sending MR becomes its child MR. In this case a MR stores information about child mobile network in its routing table.
2) If $\text{DCoA}_{\text{neighbor}} = \text{DCoA}_{\text{default}}$ and $\text{NLevel}_{\text{neighbor}} > \text{NLevel}_{\text{default}} + 1$, a sending MR becomes its descendant MR. In this case a MR ignores this RA message because a parent MR has shortest hop distance to a AR than that of the descendant MR.
3) If $\text{DCoA}_{\text{neighbor}} = \text{DCoA}_{\text{default}}$ and $\text{NLevel}_{\text{neighbor}} = \text{NLevel}_{\text{default}}$, a sending MR becomes its sibling MR. In this case a MR stores information about sibling mobile network in routing table as an alternate upstream MR that destined to the AR.
4) If $\text{DCoA}_{\text{neighbor}} \neq \text{DCoA}_{\text{default}}$, this RA message has been transmitted from a MR that has information about AR in different NEMO domain. In this case MR is multihomed. In the NEMO terminology [2], the NEMO is considered multihomed when either the NEMO is simultaneously connected to the Internet via more than one MR, or when a MR has more than one egress interface. In here we make the assumption that the NEMO has only one AR to access the Internet and is not multihomed.

Through this relation establishment, MR places the route entries in its routing table based upon the information gathered in each of the RA message received. In this way, the MR dynamically learns routes to the neighbor MRs in the NEMO domain.

2.3 Handoff Decision Algorithm for NEMO

In this section, we propose a new handoff decision algorithm called NEMO Cell Switching (NCS) to support an optimal handoff decision in multi-hop MR environments. The NCS is based on certain handoff criteria called "Internet Connectivity Strength (ICS)" and is contained the advantages of the LCS and ECS (that is, avoiding the 'ping-pong' effect and reducing the handoff delay).

In NCS algorithm, whenever a MR receives the RA messages sent by neighbors before its default router's lifetime expires, it compares the sending router and its current default router by basing on the Strength of the Internet Connectivity (ICS) to decide which is more suitable as its new default router. ICS is defined as

$$\text{Internet Connectivity Strength (ICS)} = \alpha\beta\gamma / \text{H} \qquad (1)$$

- α (Type of sending router) : If sending router is a fixed AR, a MR can access directly to the Internet. On the contrary, if sending router is a MR, it means that MR is located more than one hop away from the AR. In this case MR cannot access directly to the AR and should use its neighbor MRs to access the Internet as ad-hoc mobile networking. From this viewpoint one may say that ICS of AR is greater than that of the MR. In our proposal, the type of sending router is determined by the form of received RA message, that is, either normal RA [10] or extended RA message.

- β (State of Internet connectivity) : This parameter is used to determine whether sending router can access to the Internet. In our proposal, if a AR's advertisement is received or 'D' flag contained in the extended RA is set, one value is always assigned to the sender. If 'D' flag is unset, i.e. sending router is an isolated MR, zero value is assigned to the sender.
- γ (Domain similarity) : This parameter is used to decide handoff types, such as intra/inter domain handoff. In here 'Domain' is defined as set of all MR via the same AR to access the Internet, i.e. tree topology. After receiving RA messages, MR always checks the 'D' flag and 'Network Prefix' field contained in this message. If the 'D' flag is set and sender's DCoA matches its default router's one, the MR learns that it is still moving within the same tree domain, and needs to handle local mobility. Otherwise, the MR learns that it has entered a new tree domain, and would handle inter-domain handoff. Compared to inter-domain handoff, intra-domain handoff can reduce both handoff latency and signaling load by eliminating registration between MR and remote HA. Thus, Internet connectivity in intra-domain handoff is greater than that of inter-domain handoff.

In proportion as the parameter α, β and γ mentioned in the above rise the ICS increases. On the other hand, the 'H' parameter indicates the hop distance between sending router (or default router) and AR. Due to mobility of MRs, the topology of connection from sending router (or its default router) to AR may be quite dynamic. If the hop distance toward AR is long, that ICS can not be stable. Hence, a parameter 'H' is defined in inverse proportion to rises of ICS.

During handoff, packets addressed to MR may be lost. Packet loss will be significant, especially when the handoff process occurs frequently. This problem will degrade the communication performance. To avoid unnecessary handoff, such as 'ping-pong' problem, priority is given to a default router as described in Equation (2).

$$ICS_{default} = ((\text{RA's Lifetime}) * \alpha\beta\gamma) / H = (3 - N) * \alpha\beta\gamma / H \tag{2}$$

where the RA's lifetime is set at three times of the interval and 'N' is the number of RAs missed.

Generally MR can fail to receive a RA message from its current default router. In our proposal MR keeps a counter that counts the number of RAs missed for its default router. The count is incremented on the expiry of RA interval. If the missing RA count reaches doubles, ICS of the current default router is the same as Equation (1). In the case of three consecutive missing RAs, MR concludes that its current default router as unreachable because the $ICS_{default}$ = zero. This is same as LCS algorithm.

An Equation (1) also can define as shown in the below according to the type of sending router.

$$ICS_{sender} = \alpha\gamma, \text{ if sending router is a fixed AR or}$$

$$ICS_{sender} = \beta\gamma / H, \text{ if sending router is a MR or}$$

$$ICS_{sender} = \text{zero} \text{ , if sending router is an isolated MR}$$

As described in the above, after receiving a new RA message, MR calculates the ICS of its default router and sending router, respectively. If the ICS of the sending router is greater than that of its default router as Equation (3), a handoff will be performed.

$$ICS_{default} < ICS_{sender} \text{ ; handoff execution} \tag{3}$$

3 Performance Evaluation

To show how well NCS algorithm performs, we compare the performance of the three handoff decision algorithms, i.e. ECS, LCS and NCS. The simulation was based on the Network Simulator (NS-2) [11] and Mobiwan [12] developed by the Motorola. The scenario we have studied includes 32 mobile networks that are placed randomly over a rectangular (800m x 800m) flat space for 1,000 seconds of simulated time, and are connected to the ARs with a hierarchical tree structure, i.e., tree-based NEMO scheme. To simulate real traffic, we set up the CN as a traffic source of a Constant Bit Rate (CBR) source over a User Datagram Protocol (UDP), producing fixed length packets of 1000 bytes each every 1 second. A moving MR1 acts as a sink receiving packets from CN.

Table 1. Parameters for NCS Algorithm

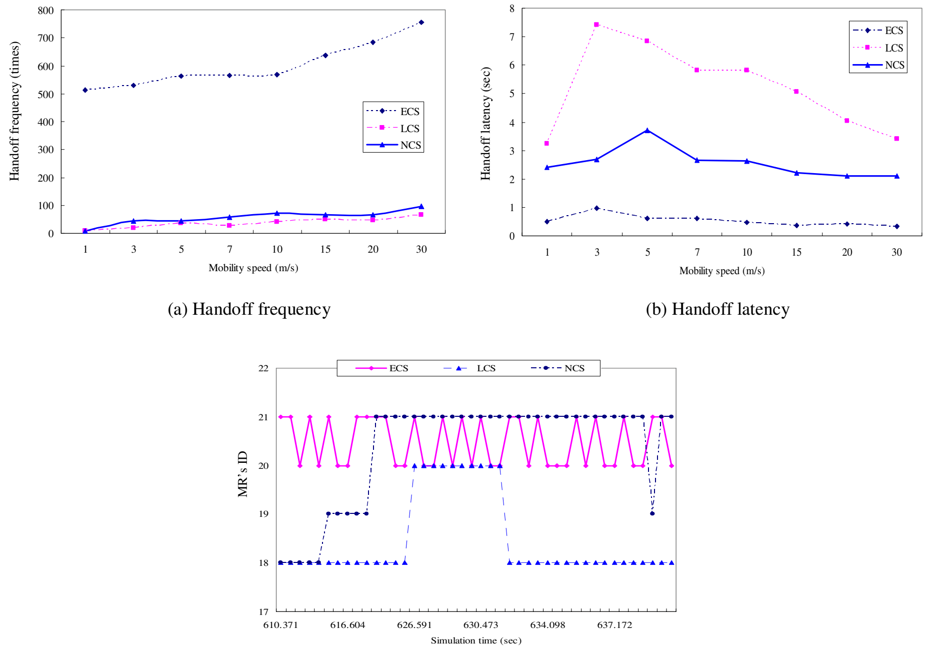
Parameter	Description	Values used in simulation
N	Number of RAs missed	0 ~ 3
H	Hop count from its default router (or a sending router) to AR	1 ~
α	Type of sender (or its default) router	2 (AR) 1 (MR)
β	State of Internet connectivity	1 (AR or 'D' flag is set) 0 (an isolated MR)
γ	Domain similarity	2 (same domain) 1 (different domain)

Since the handoff decision algorithm concerns the communication between the AR (or MR) and MRs, we have only focused on the wireless part of the scenario. In order to simulate handoff decision efficiently, mobility speed of a moving MR1's neighbors (MR2 ~ MR32) set to zero, i.e., have kept fixed with 0 m/s during all simulations, and propagation delay is ignored. The RAs period for these MRs is 1 second, but the RAs from these MRs are not synchronized.

When comparing the performance of the handoff decision algorithms, we use two metrics, i.e. handoff frequency and handoff frequency, in terms of the mobility speed, the number of MRs and RA intervals, respectively. In this simulation, the main parameters used for NCS are shown in Table 1.

Mobility Speed and Handoff Performance

Fig. 2 shows simulation results for handoff performance when the mobility speed of a moving MR is varied. In Fig.2 (a), ECS has higher handoff frequency than other two



(a) Handoff frequency

(b) Handoff latency

(c) Ping-pong handoffs

Fig. 2. Mobility Speed and Handoff Performance

algorithms. It is believed that a handoff initiates immediately upon learning a new network prefix. Unlike ECS, LCS and NCS perform well though the frequency increases slightly as the mobility speed increases. This is because MR does not initiate a handoff until the current point of attachment is confirmed to be unreachable (LCS) or until the ICS of a new neighbor is greater than that of its current router (NCS).

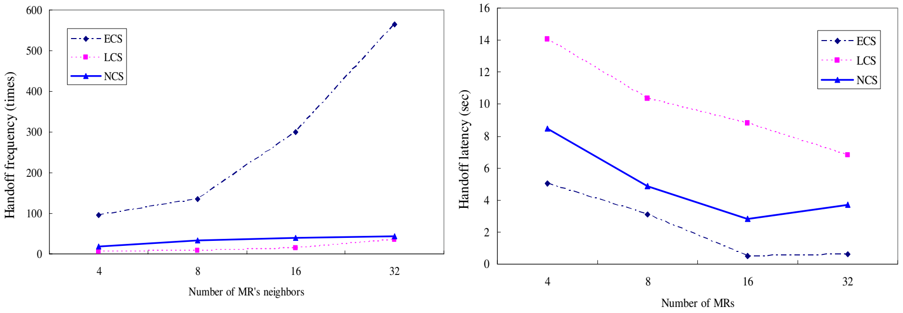
The handoff latency of three algorithms shows in Fig. 2 (b). In Fig. 2(b), the handoff latencies associated with LCS and NCS are high than that of ECS. This is because it takes some time to determine whether to perform a handoff (NCS) or to wait until its current router is confirmed to be unreachable (LCS). However, compared to the LCS, the NCS offers lower handoff latency. It is because MR in NCS initiates a handoff whenever the Internet connectivity of a new neighbor is greater than that of its current router. From the results, the handoff latency of ECS is decreased slightly for the speed of the MR. This is because that the possibility of receiving a new RA increases as the mobility speed increases, and a handoff in ECS initiates immediately upon receiving a new RA.

In Fig. 2(b), ECS may offer better handoff latency performance than both LCS and NCS, but it will result in unnecessary handoff to the other neighbor MR as shown in Fig. 2 (c). We see that in the case of ECS a handoff is performed repeatedly between MR20 and MR21. That is, the ping-pong problem happens frequently in ECS, unlike LCS and NCS.

Number of MRs and Handoff Performance

Fig. 3 is to evaluate the performance of three handoff decision algorithms as a function of the number of MRs, which is located within NEMO domain. In Fig.3 (a), Both LCS and NCS exhibit the lower handoff occurrence, i.e. the handoff frequency is slightly increased as the number of MRs increases. This is because that the decision whether or not to perform a handoff is decided by a reactive handoff initiation method. That is, a moving MR does not initiate a handoff until the current network becomes unavailable (LCS) or until the ICS toward a new network is greater that of its current network (NCS). Therefore, we can see that handoff frequency of both LCS and NCS does not influenced greatly by the number of MRs. On the contrary, as the number of MRs increase, the handoff frequency of ECS increases rapidly. This is due to two main reasons:

- ECS is to change network as soon as a new network is discovered. Thus, the possibility of handoff occurrence is high in NEMO domain where many MRs are deployed.
- If many MRs are deployed in close vicinity, a moving MR can be located in wireless environments with two or more overlapping RAs. This will result in the so called “ping-pong” effect which makes the MR switches between the neighbors within coverage.



(a) Handoff frequency

(b) Handoff latency

Fig. 3. Handoff frequency vs. Number of MRs

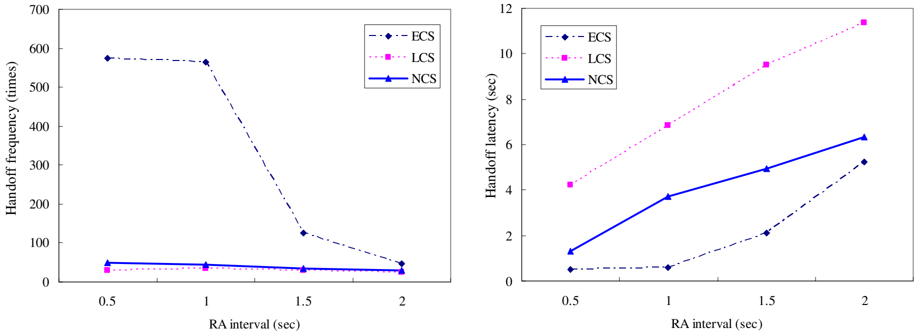
Fig. 3 (b) illustrates the handoff latency. On the whole, as the number of MRs increase, the latency decreases. This is because that it takes long time to find a new default router at the small number of MRs after/before the current router’s lifetime is expired. When the number of MR goes over 16, the latency begins to vary constantly. As expected, since LCS is never initiated before the current network is declared unreachable, handoff latency of LCS is high than that of both ECS and NCS. Compared ECS, NCS handoff latency is high slightly since NCS evaluates a set of handoff criteria and determines which MR provides the best performance from a moving MR point of view.

RA Interval and Handoff Performance

As NEMO uses the reception of RAs to discover new networks, the interval between sending RAs can affect the time it takes to discover new networks. Generally, frequent

RA is helpful for movement detection and it causes shorter handoff latency [13, 14]. However, processing frequent RA is significant overhead for MRs in the view of networks.

Fig. 4 shows the handoff performance of a moving MR when the interval of the MR's RA is varied. As expected, when the interval between RAs is 0.5 seconds in Fig. 4 (a), ECS exhibits the higher handoff occurrence. This is because that fast RA interval can directly affect the behavior of ECS and a moving MR performs unnecessary handoff, i.e. ping-pong, continuously at overlapped zone between the neighbors. When the interval of RAs goes over a certain value, the handoff frequency begins to decrease rapidly. In both LCS and NCS, handoff occurs constantly between 0.5 seconds and 2 seconds. This is because that both LCS and NCS handoff initiation methods are not influenced by the interval between receiving unsolicited RAs and by the reachabilty current network.



(a) Handoff frequency

(b) Handoff latency

Fig. 4. RA intervals vs. Handoff performance

Fig. 4 (b) shows the average of handoff latency when the interval of RA is varied. The simulation results say that all the handoff decision methods are increased rapidly as the interval of RAs increases. This is because that the time of handoff detection which triggers a handoff to a new network increases according to the interval increasing. Compared to the ECS, LCS and NCS offer lower handoff latency.

4 Conclusion

In this paper, the mechanisms for optimal handoff decision under multi-hop NEMO environments, not only supporting Internet connectivity and reducing the routing overhead, but also avoiding the 'ping-pong effect' and frequent handoffs, are proposed.

First, extending the MIPv6's RA message is suggested to discover its default MR which has the Internet connectivity, and we then described the optimized handoff decision procedure to decide an optimal handoff under multi-hop NEMO environments. This procedure is partitioned into four phases: 1) Check the reachability, 2) Neighbor router's type decision, 3) Handoff decision and execution, and 4) Relation decision between MRs. Next, a new handoff decision algorithm called NEMO Cell Switching (NCS) is proposed to support an optimal handoff decision in multi-hop MR environments. The NCS is based on certain handoff criteria called "Internet Connectivity Strength (ICS)".

To evaluate the performance of the handoff decision mechanisms proposed, we have performed a series of simulations. The simulation was based on the Network Simulator (NS-2) and MobiWAN developed by Motorola. The simulations have been executed under various simulation environments taking into account mobility speed, the number of mobile networks, RA intervals. The simulation results show that the proposed approach, i.e., NCS algorithms, outperforms the existing approaches in most cases. Thus, we believe that the work presented is an important step towards supporting optimal handoff.

Reference

[1] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004

[2] Thierry Ernst, Hong-Yon Lach, "Network Mobility Support Terminology", < draft-ietf-nemo-terminology-01.txt>, Feb. 2005

[3] V. Devarapalli, R. Wakikawa and P. Thubert, "Nemo Basic Support Protocol", RFC 3963, Jan. 2005

[4] T.Ernst,, Keisuke Uehara and Koshiro Mitsuya "Network Mobility from the InternetCAR perspective", Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference, pp. 19 – 25, March 2003

[5] IETF NEMO WG, http : //www.mobilenetworks.org/nemo

[6] N.Blefari-Melazzi, M.Femminelle, F.Pugini, "Movement detection in IP heterogeneous wireless networks", Personal Mobile Communications Conference, 2003. 5th European (Conf. Publ. No. 492), pp. 121 – 125, April 2003

[7] N. Blefari-Melazzi, M. Femminella, F. Pugini, "A robust and flexible solution for Movement Detection in Wireless IP Networks: Enhanced Lazy Cell Switching", Proceedings of IST Mobile & Wireless Telecommunications Summit 2002, June 2002

[8] Daley, G., Pentland, B., Nelson. R, "Movement detection optimizations in mobile IPv6", ICON2003, The 11th IEEE International Conference, pp. 687 – 692, Sept. 2003

[9] H. Cho, E. K. Paik, "Hierarchical Mobile Router Advertisement for nested mobile networks", < draft-cho-nemo-hmra-00.txt>, work in progress, Jan. 2004

[10] T. Narten, E. Nordmark and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, Dec. 2004

[11] The VINT Project, The UCB/LBNL/VINT Network Simulator-ns (version2), http://www.isi.edu/nsnam/ns.

[12] MobiWAN: NS-2 extensions to study mobility in Wide-Area IPv6 Networks, http://www.inrialpes.fr/planete/pub/mobiwan/

[13] Daley G., Pentland B., Nelson R., "Movement detection optimizations in mobile IPv6", Networks, 2003. ICON2003. The 11th IEEE International Conference, pp. 687 – 692, Sept. 2003

[14] Daley G., Pentland B., Nelson R., "Effects of fast routers advertisement on mobile IPv6 handovers", Computers and Communication, 2003. (ISCC 2003), Eighth IEEE International Symposium, pp. 557 – 562, 2003

Fast Re-authentication for Handovers in Wireless Communication Networks

Ralf Wienzek and Rajendra Persaud

Chair of Informatik 4, RWTH Aachen University, Aachen, Germany
{wienzek, persaud}@i4.informatik.rwth-aachen.de

Abstract. The evolution of wireless access technologies and the capabilities of today's mobile devices lead to an increasing demand of communication bandwidth. More and more packet-switched wireless access networks like Wireless Local Area Networks (WLANs) and networks based on Worldwide Interoperability for Microwave Access (WiMAX) are publicly available and operated by different providers. In order to achieve a high network coverage isolated access network providers are supposed to co-operate and to support handovers for users from access networks belonging to the same core network. Efficient authentication mechanisms are required that on the one hand exclude unauthorized users from the network and on the other hand support seamless handovers across access network boundaries. We propose a ticket-based fast re-authentication scheme that is independent from the actual authentication method and that only slightly modifies well-established standards like the Extensible Authentication Protocol (EAP) and the Remote Authentication Dial In User Service (RADIUS). As it is network technology independent, it in principle also allows fast handovers across different access network technologies.

Keywords: Wireless Networks, Authentication, Handover.

1 Introduction

Mobility is one of the main incentives for the development of wireless network technologies such as Wireless Local Area Networks (WLANs) based on IEEE 802.11 or Worldwide Interoperability for Microwave Access (WiMAX) based on IEEE 802.16. In general, a wireless network is subdivided into one or more access networks and a core network that do not need to belong to the same network operator. An access network consists of link-layer (L2) devices, whereas a core network consists of network-layer (L3) devices. The device the link layer of a mobile device attaches to is called L2 Point of Attachment (PoA) (Access Point (AP) in WLANs, Base Station (BS) in WiMAX networks), the device the network layer of a mobile device attaches to is called L3 PoA (Access Router (AR) in WLANs, Access Services Network Gateway (ASN-GW) in WiMAX networks).

In both technologies, access network mobility is in general provided by the technology itself, i.e. a handover (HO) between PoAs belonging to the same

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 556–567, 2006.

access network does not require any higher-layer mobility solution. Core network mobility, i.e. a HO across access network boundaries, is in general based on the Internet Protocol (IP) or on Multi-Protocol Label Switching (MPLS) and goes along with a reconfiguration in the core network in order to redirect packets to the new PoA.

Mobility is a user-specific network service that has to be secured from attackers and therefore requires an authentication mechanism. In order to support real-time packet-switched applications such as Voice over IP (VoIP), the HO from an old PoA to a new PoA and the associated re-authentication with the new PoA has to be as fast as possible. The objective of this paper is to provide a fast re-authentication mechanism to support core network mobility.

In general, the mobile device authenticates with an Authentication, Authorization and Accounting (AAA) server in the core network. Also, authentication is reasonably performed before obtaining network-layer connectivity, i.e. before the assignment of an IP address. Therefore, an intermediate device is necessary to handle the authentication between the mobile device and the AAA server. In WLANs based on IEEE 802.11i this is the L2 PoA. In WLANs based on IEEE 802.11 only, the L2 PoA cannot act as intermediate device so that the L3 PoA has to be used.

During HO, a HO notification message has to be sent to the core network (e.g. by exploiting the Dynamic Host Configuration Protocol [1], which is an extensible protocol used for configuration purposes). We propose a predictive HO solution, i.e. the HO notification is sent to the L3 PoA the mobile device is attached to before it moves to the new access network. The advantage compared to reactive HO solutions in which HO notifications are sent to the new L3 PoA is that the disassociation from the old access network and the transfer of configuration information from the old L3 PoA to the new L3 PoA can be done in parallel.

In order to accelerate the authentication process during HO we propose a ticket mechanism providing a fast re-authentication of mobile nodes with the target access network. We define an additional RADIUS attribute and use the optional data field of EAP-Identity-Messages.

The rest of the paper is organized as follows: In Sect. 2 we summarize EAP-TLS with RADIUS as an example scenario to which the re-authentication scheme can be applied. In Sect. 3 we define the attacker model and describe our scheme in detail. In Sect. 4 required adaptations to apply the scheme to IEEE 802.11i like environments are given. Performance and security issues are discussed in Sect. 5 and the paper closes with some conclusions in Sect. 6.

2 Authentication Schemes

Mobile nodes (MN) that want to use core network services are required to authenticate themselves with the core network. When a MN enters an access network it first attaches itself to the L2 PoA. The authentication procedure with the core network is initiated either by directly contacting the L3 PoA managing that access network or in combination with an authentication required for associating with the L2 PoA. An example for the first case is an IEEE 802.11 WLAN. The L3 PoA can be contacted e.g. by assigning a temporary IP address with restricted access rights to the MN as proposed in [2] or by establishing a Point-to-Point connection using the Point-to-Point Protocol over Ethernet (PPPoE) [3]. An example for the second scenario is an IEEE 802.11i AP that authenticates a MN with the help of an AAA server. Our scheme covers both scenarios. We define it for the first, simpler scenario in detail, and describe necessary adaptations for IEEE 802.11i environments in Sect. 4.

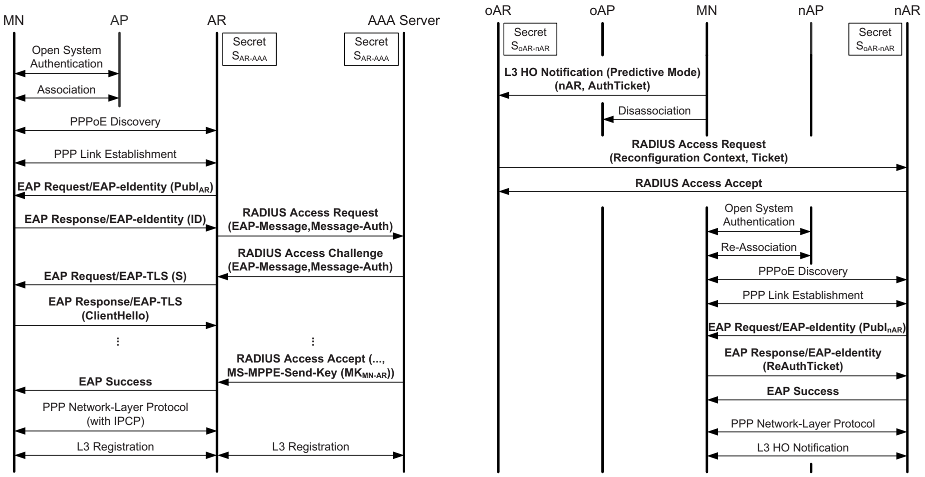
For security and maintenance reasons, the deposited authentication credentials used for authenticating a user are not directly available to the L3 PoA but rather centrally stored on an AAA server. The L3 PoA relays the authentication messages of both the MN and the AAA server and, at the end, is informed by the AAA server whether it should allow the MN to access the network or not. Message authentication for L3 signalling traffic is implemented by using key material established when the MN and the network (represented by the AAA server) authenticate each other. As the L3 PoA has to be able to create and verify authenticated messages, the AAA server transmits the necessary key material to the L3 PoA over a secured channel.

The Extensible Authentication Protocol (EAP) [4] and the Remote Authentication Dial In User Service (RADIUS) [5] are two widely-used protocols for authentication purposes. EAP provides a flexible framework allowing arbitrary authentication mechanisms. In our scenario it is used for the communication between the MN and the AAA server in which the L3 PoA acts as authenticator in pass-through mode. RADIUS is used to transport information between the AAA server and the L3 PoA. As an example scenario, to which our re-authentication scheme can be applied, in the following the authentication procedure based on Transport Layer Security (TLS) [6] is briefly described for an IEEE 802.11 environment.

The PPP EAP TLS Authentication Protocol as defined in [7] provides mutual authentication between an EAP client (here: the MN) and an EAP server (here: the AAA server) and allows to establish key material to be used for a subsequent secure communication. The L3 PoA (here: the AR) acts as the EAP authenticator. In [7] a specific EAP method called EAP-TLS is standardized that defines the transport of TLS messages within EAP messages.

The message flow for the registration of a MN with an access network is depicted in Fig. 1 (a). After associating with the AP, a link between the MN and the AR is established, e.g. by running PPPoE.

An EAP-Identity-Request is issued by the AR that is answered by the client with an EAP-Identity-Response containing an identifier. According to [4], EAP-Identity-Requests can optionally contain data to be displayed to the user and, additionally, data to be used as initialization of subsequent authentication methods. We exploit this option to implement our fast re-authentication procedure. Therefore, in Fig. 1 (a) the EAP-Identity request is already denoted by EAP-eIdentity.


MN
AP
AR
AAA Server
Secret $S_{\mathrm{AR-AAA}}$
Secret $S_{\mathrm{AR-AAA}}$
Open System Authentication
Association
PPPoE Discovery
PPP Link Establishment
EAP Request/EAP-eIdentity ($\mathrm{Publ}_{\mathrm{AR}}$)
EAP Response/EAP-eIdentity (ID)
RADIUS Access Request (EAP-Message,Message-Auth)
RADIUS Access Challenge (EAP-Message,Message-Auth)
EAP Request/EAP-TLS (S)
EAP Response/EAP-TLS (ClientHello)
⋮
⋮
RADIUS Access Accept (..., MS-MPPE-Send-Key ($\mathrm{MK}_{\mathrm{MN-AR}}$))
EAP Success
PPP Network-Layer Protocol (with IPCP)
L3 Registration
L3 Registration
oAR
oAP
MN
nAP
nAR
Secret $S_{\mathrm{oAR-nAR}}$
Secret $S_{\mathrm{oAR-nAR}}$
L3 HO Notification (Predictive Mode) (nAR, AuthTicket)
Disassociation
RADIUS Access Request (Reconfiguration Context, Ticket)
RADIUS Access Accept
Open System Authentication
Re-Association
PPPoE Discovery
PPP Link Establishment
EAP Request/EAP-eIdentity ($\mathrm{Publ}_{\mathrm{nAR}}$)
EAP Response/EAP-eIdentity (ReAuthTicket)
EAP Success
PPP Network-Layer Protocol
L3 HO Notification

(a) Startup (b) Handover

Fig. 1. Startup and Handover for IEEE-802.11-based access networks using EAP-eIdentity and EAP-TLS authentication (authentication-relevant messages in bold)

The additionally transported public key $Publ_{\mathrm{AR}}$ is an optional parameter and is ignored during startup.

As the AR has no access to the credentials for authenticating a MN, it acts as a RADIUS client and contacts the AAA server acting as RADIUS server. In [8] two attributes (EAP-Message and Message-Authenticator) are introduced to support an authenticated EAP message transport within RADIUS packets. The authentication is based on a secret key $S_{\mathrm{AR-AAA}}$ that has to be established between the AR and the AAA server by other means.

The actual EAP TLS authentication procedure uses a public key infrastructure based on certificates. Within three round trips, both the MN and the AAA server submit a random number, their public key along with a certificate chain proving the key's authenticity, and signatures computed over the messages exchanged so far to authenticate themselves. Furthermore, the MN sends a premaster secret, i.e. a random number of appropriate length, encrypted with the server's public key. The master key, i.e. the shared key material between the AAA server and the MN, is computed from these random numbers and the premaster secret.

If the EAP TLS authentication procedure is successfully finished the AAA server will send a RADIUS-Access-Accept packet to inform the AR that the MN is authenticated and can be given access to the network. This packet also contains the established master key encapsulated e.g. into MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes defined in [9]. In order to particularly protect this message, the whole communication between the AR and the AAA server should be protected by e.g. IPsec. Finally, the AR issues a EAP-Success-Message to the MN which then gets configured and registered with the core network.

3 Fast Re-authentication

As indicated in Fig. 1 (a) the conventional authentication procedure consists of several round trips between the MN and the AAA server. Furthermore, when using public key certificate chains the verification of certificates can become a time-consuming task. As packet-switched multimedia applications like video streaming or VoIP benefit from a minimal link downtime during HO, the re-authentication scheme has to be as fast as possible.

On the other hand, malicious nodes have to be prevented from misusing the re-authentication method in any way. Therefore, the new key material between a MN and its new AR should be as strong as the old one.

3.1 Attacker Model

Attackers the network has to be protected from are assumed to be mobile stations located within communication range of a MN and/or AP and trying to retrieve valuable information by eavesdropping the channel and injecting forged messages. We only consider the signalling traffic between a MN and its AR and do not address security issues on the transport or application layer as appropriate measures are available (e.g. IPsec, VPN, SSL, etc.). Furthermore, we assume that the fixed infrastructure components like ARs and AAA servers are not compromised, behave according to the protocol, and do not collude with malicious MNs. AAA servers accept RADIUS requests only from registered ARs and the communication between an AR and its AAA server is encrypted and authenticated. This can be achieved by configuring both peers with a shared secret key or by using mutually authenticated IPsec channels.

In our attacker model the attacker has the following capabilities:

- **Eavesdropping:** The attacker is able to retrieve all data packets sent over the air interface. If an encryption mechanism is in use, the attacker can only read the packet's content if it is in possession of the decryption key.
- **Forging:** The attacker is able to compile arbitrary packets and has full control over the packets' headers and data parts.
- **Simulating infrastructure devices and network services:** An attacker is able to simulate wireless access points, base stations, service gateways, etc. as long as the needed information is publicly available or has been gathered in previous attacks.
- **No resource limitations:** The attacker does not suffer from power or computational limitations that battery driven devices normally have to deal with.

Without an appropriate protection of the signalling traffic between a MN and its AR the above defined attacker would have several possibilities to attack the network. It could send forged HO requests in the name of other mobile nodes to the AR and may disconnect that node from the network. Furthermore, it could request core network services in the name of other nodes and by this gather sensitive information or trigger reconfiguration procedures. If the attacker intercepts a regular HO request it could try to connect to the destination network, pretend to be the node that requested the HO and get unauthorized network access. As the attacker is able to simulate infrastructure components, it could also pose as the new AP/AR, trick the mobile node to connect to it, and perform at least some form of denial of service or at worst gather sensitive information. As HOs between access networks of different providers are supported it might be interesting for an old provider to further eavesdrop the signalling communication between the MN and the new provider after the HO.

Consequently, when a mobile node M performs a HO from an access network managed by the access router OA into an access network managed by NA, several requirements have to be fulfilled by the underlying re-authentication scheme:

– NA has to recognize M as being authorized to access the network. Furthermore, in order to preserve higher level security associations based on constant addresses, M should be assigned the same network address in the new access network as it had been configured with in the old one.
– The new key material for a secured channel between M and NA has to be exchanged in such a way that (1) NA cannot determine the key used between M and OA and (2) OA cannot derive the new key used by M and NA.
– Unauthorized nodes are not able to exploit the protocol to get network access or gather any other advantages.
– The delay caused by the authentication procedure has to be low.

3.2 Re-authentication Scheme

For our re-authentication scheme we assume that a secret key $S_{\mathrm{MN-oAR}}$ has been established between the MN and the old AR that can be used to exchange secured messages. Generally, this key is exchanged during startup. Furthermore, we assume the existence of a protected channel between the old AR and the new AR based on the key $S_{\mathrm{oAR-nAR}}$ established via manual configuration or by using e.g. the Internet Key Exchange protocol. In practice, an AR will probably not have very many neighbor ARs so that a manual configuration might be feasible. From the security point of view we have authenticated relationships between the MN and the old AR and between the old AR and the new AR. Our goal is to establish such a relationship between the MN and the new AR.

The message sequence of our fast re-authentication scheme is depicted in Fig. 1 (b). The MN sends a HO notification message to its current AR. It contains information about the destination access network and an authentication ticket *AuthTicket* consisting of a sequence number used to prevent replay attacks and the actual authentication information $(\mathit{MN\_ID},\ \mathit{nMK})_{SK}$ to be forwarded to the new AR.

$$\mathit{AuthTicket} := \{\mathit{SeqNo}, (\mathit{MN\_ID}, \mathit{nMK})_{SK}\}_{S_{\mathrm{MN-oAR}}} \tag{1}$$

nMK is the new, randomly chosen master key between the MN and the new AR. *MN_ID* is the ID of the MN and is used to bind *nMK* to that particular MN. In order to disguise *nMK* from the old AR the authentication information is encrypted with the randomly chosen key *SK*. The *AuthTicket* is encrypted and authenticated by using the current session key $S_{\mathrm{MN-oAR}}$.

When the old AR receives the *AuthTicket* it verifies the sequence number and forwards a *Ticket*, as an attribute of a RADIUS-Access-Request packet, over the previously established secured channel to the new AR.

$$Ticket := \{MN\_ID, (MN\_ID, nMK)_{SK}\}_{S_{\mathrm{oAR-nAR}}} \tag{2}$$

The old AR prepends the *MN_ID* of the MN with which it shares $S_{\mathrm{MN-oAR}}$ to the authentication information. By this, the new AR later can verify that the *MN_ID* provided by the MN is the same as the one the MN has used to authenticate with the old AR. Along with the *Ticket* a reconfiguration context is submitted, i.e. information necessary to reconfigure the MN in the new access network. The new AR stores the *Ticket*, starts a timer, and waits for the MN to enter the network and request a fast re-authentication. If the MN does not show up before the timer runs out, the *Ticket* will be deleted. In order to re-authenticate itself to the new access network, the MN has to prove its knowledge of *nMK* stored in the corresponding *Ticket*. Furthermore, the new AR has to be enabled to extract *nMK* from the *Ticket* while preventing the old AR from doing the same.

After the conventional open system authentication with the new AP and the link establishment with the new AR, e.g. via PPPoE, the new AR requests the MN to authenticate itself. At this point in time, the new AR does not know, whether it has to handle a conventional or a fast re-authentication. As in the startup example, the new AR sends an extended EAP-Identity-Request called *EAP-eIdentity* that contains its public key $Publ_{\mathrm{nAR}}$. A MN can answer to this request with a conventional EAP-Identity-Response, thus initiating the conventional authentication scheme as described in Sect. 2. Alternatively, it can provide the re-authentication ticket *ReAuthTicket* and get authenticated immediately[1].

$$ReAuthTicket := \{MN\_ID, oAR, (SK)_{Publ_{\mathrm{nAR}}}, HMAC_{nMK}(Publ_{\mathrm{nAR}})\} \tag{3}$$

The *MN_ID* in the *ReAuthTicket* aims at identifying the correct *Ticket* from the list of currently pending tickets at the new AR. The address of the old AR *oAR* enables the new AR to contact the old AR in case that the correct *Ticket* has not yet been received. The secret key *SK* used to encrypt *nMK* in (2) is encrypted with $Publ_{\mathrm{nAR}}$. The old AR is therefore prevented from getting knowledge of *SK* by eavesdropping the *ReAuthTicket*. The last component is a hashed message authentication code (HMAC) computed over $Publ_{\mathrm{nAR}}$ and seeded with *nMK*. On the one hand it is generated to prove knowledge of *nMK* and on the other hand to indicate that it is generated to answer an EAP-eIdentity-Request with $Publ_{\mathrm{nAR}}$. The new AR uses its private key to extract *SK* from (3) and uses *SK* to decrypt the new master key *nMK* and the encrypted *MN_ID* from (2). By verifying that both *MN_IDs* from (2) match, it is ensured that the MN has

[1] Although the EAP standard [4] states to send EAP-Success-Messages "after completion of an EAP authentication method (Type 4 or greater)", sending an EAP-Success-Message already after receiving an EAP-Identity-Response (Type 1) is not explicitly forbidden.

transmitted the *MN_ID* it has used to authenticate itself with the core network. A successful verification of the HMAC provided in (3) proves that the issuer of the *ReAuthTicket* knows *nMK*. As the *Ticket* comes from a trusted peer (the old AR) to which the MN had authenticated itself, the MN is successfully authenticated with the new AR and the IP address stored in the reconfiguration context can be assigned to the MN with ID *MN_ID*. Finally, the core network is informed about the performed HO operation and the new AR can delete the used ticket from its list of pending tickets.

Although intended to be used in predictive mode it is also possible to apply the re-authentication scheme in reactive mode. For this, the MN would have to transmit the *AuthTicket* (1) along with the *ReAuthTicket* (3) in the EAP-Identity-Response. The new AR would transmit the *AuthTicket* to the old AR which could then transmit the reconfiguration context along with the *Ticket* (2) to the new AR. Afterwards, the verification of the *ReAuthTicket* can be performed as described above.

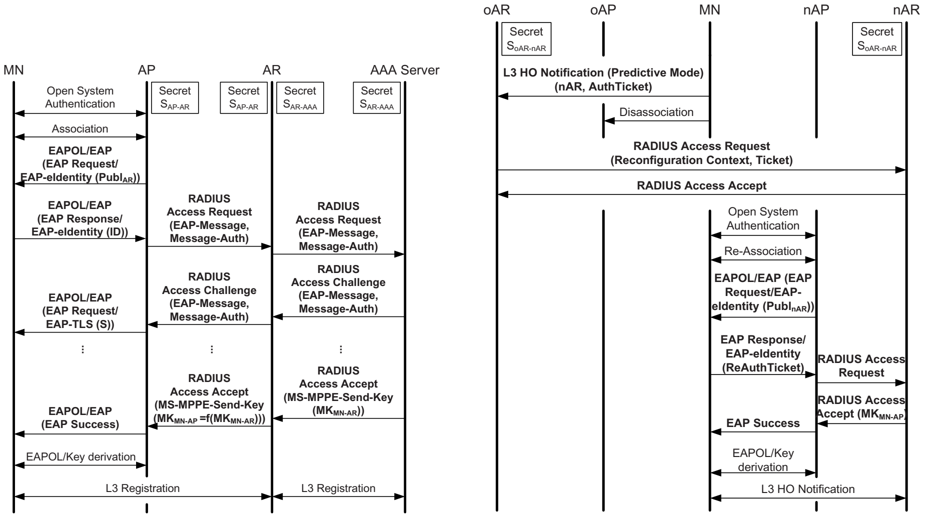
In the case the MN decides to re-attach itself to the old AR instead of the new AR, the fast re-authentication can be applied analogously, with the special case that old AR and new AR are identical.

4 Application to IEEE 802.11i Environments

In IEEE 802.11i environments an AP itself runs authentication procedures. It can be configured to act as an EAP authenticator and to negotiate a master key between a MN and itself by using the mechanisms described in Sect. 2. As the AR is no longer actively involved in this process a key between the MN and the AR used to protect L3 signalling traffic is not established.

As a solution we propose to configure the AP to use its AR as RADIUS server. With exception of the final RADIUS-Access-Accept packet, the AR acts like a RADIUS proxy by relaying the RADIUS packets between AAA server and AP (cf. Fig. 2 (a)). The MN and the AAA server negotiate a master key $MK_{\mathrm{MN-AR}}$ by running EAP-TLS and the AAA server transmits $MK_{\mathrm{MN-AR}}$ in its final RADIUS-Access-Accept packet to the AR. The AR uses $MK_{\mathrm{MN-AR}}$ on the one hand to derive session keys for a secured communication with the MN and on the other hand to derive the master key $MK_{\mathrm{MN-AP}}$ for the AP with a publicly-known one-way function. $MK_{\mathrm{MN-AP}}$ is transmitted to the AP in the corresponding RADIUS-Access-Accept packet. On receipt of the EAP-Success-Message from the AP, the MN also computes $MK_{\mathrm{MN-AP}}$ from $MK_{\mathrm{MN-AR}}$.

Analogously, the fast re-authentication is implemented for this scenario (cf. Fig. 2 (b)). The ticket is created in the same manner as described before and transmitted via the old AR to the new AR. After a successful L2 re-association the re-authentication is slightly different from the one illustrated in Fig. 1 (b). The new AP has to start the authentication procedure by sending the initial EAP-eIdentity-Request containing its AR's public key $PublnAR$...


MN
AP
AR
AAA Server
Secret S_{AP-AR}
Secret S_{AP-AR}
Secret S_{AR-AAA}
Secret S_{AR-AAA}
Open System Authentication
Association
EAPOL/EAP (EAP Request/ EAP-eIdentity ($PubI_{AR}$))
EAPOL/EAP (EAP Response/ EAP-eIdentity (ID))
RADIUS Access Request (EAP-Message, Message-Auth)
RADIUS Access Request (EAP-Message, Message-Auth)
RADIUS Access Challenge (EAP-Message, Message-Auth)
RADIUS Access Challenge (EAP-Message, Message-Auth)
EAPOL/EAP (EAP Request/ EAP-TLS (S))
RADIUS Access Accept (MS-MPPE-Send-Key (MK_{MN-AP} =f(MK_{MN-AR})))
RADIUS Access Accept (MS-MPPE-Send-Key (MK_{MN-AR}))
EAPOL/EAP (EAP Success)
EAPOL/Key derivation
L3 Registration
L3 Registration
oAR
oAP
MN
nAP
nAR
Secret $S_{oAR-nAR}$
Secret $S_{oAR-nAR}$
L3 HO Notification (Predictive Mode) (nAR, AuthTicket)
Disassociation
RADIUS Access Request (Reconfiguration Context, Ticket)
RADIUS Access Accept
Open System Authentication
Re-Association
EAPOL/EAP (EAP Request/EAP-eIdentity ($PubI_{nAR}$))
EAP Response/ EAP-eIdentity (ReAuthTicket)
RADIUS Access Request
RADIUS Access Accept (MK_{MN-AP})
EAP Success
EAPOL/Key derivation
L3 HO Notification

(a) Startup

(b) Handover

Fig. 2. Startup and Handover for IEEE-802.11i-based access networks using EAP-eIdentity and EAP-TLS authentication

AP forwards the response via RADIUS to the new AR which verifies the re-authentication ticket, extracts the new master key *nMK*, derives the master key $MK_{\mathrm{MN-AP}}$ to be used between the MN and the new AP, and transmits $MK_{\mathrm{MN-AP}}$ encapsulated into a RADIUS-Access-Accept packet to the new AP. The MN derives $MK_{\mathrm{MN-AP}}$ from *nMK* in the same way.

5 Evaluation

In this section the fast re-authentication scheme is evaluated according to the requirements of Sect. 3.1.

5.1 Security Evaluation

The security evaluation of the proposed re-authentication scheme is based on the assumptions that the underlying cryptographic functions are secure and that random numbers are not predictable. Furthermore, we assume that the EAP TLS authentication with RADIUS as described in Sect. 2 is secure.

On the one hand it has to be shown that authorized nodes – clients as well as servers – are able to successfully perform a fast re-authentication. On the other hand any attacker with capabilities as described in Sect. 3.1 must be prevented from misusing the scheme to get any advantages.

Authorized nodes are able to re-authenticate each other: A node that has been granted access to the old access network has to be granted access to the new one and it can be configured with the same network address as before.

Furthermore, any AR that is supposed to be the new AR for a MN, has to have the ability to verify an authorized request and to prove to the MN that it is the component it claims to be. Obviously, a MN is able to create a matching pair of *AuthTicket* (1) and *ReAuthTicket* (3) and the new AR has the ability to verify *ReAuthTicket* with information from *Ticket* (2) as described in Sect. 3.2. The configuration context transmitted along with the *Ticket* allows the new AR to configure the MN with the same network address as before. By using key material derived from *nMK* to authenticate subsequent signalling messages, the AR can prove its eligibility to the MN. A minor drawback of our scheme is the fact that the new AR is not authenticated until it sends the first signalling message that has to be protected. A challenge/response mechanism could resolve this issue but it would mean an additional round trip between MN and AR or a modification of the final EAP-Success-Message.

Attack prevention: At first we prove a lemma from which the other security properties can easily be derived.

Lemma 1. *If the above-mentioned assumptions are fulfilled an attacker with the capabilities as described in Sect. 3.1 is not able to retrieve nMK from the fast re-authentication process.*

Proof. The value *nMK* is randomly chosen by the MN. As neither the old AR nor the new AR nor the MN are compromised and random numbers are unpredictable, the only chance for an attacker to gain knowledge of *nMK* is to derive it from messages. By passively eavesdropping the air interface, the attacker can obtain *AuthTicket* (1) and *ReAuthTicket* (3). Furthermore, it can get knowledge of *SK* by actively forging an initial EAP-eIdentity-Request with its own public key $Publ_{\mathrm{A}}$. If the public key is not enriched with a certificate – which is, due to performance reasons, assumed to be the case – the MN will issue the re-authentication ticket based on that public key and therefore reveal *SK* to the attacker.

As the underlying cryptographic functions are secure, $(MN\_ID, nMK)_{SK}$ remains unintelligible to the attacker (it is encrypted with $S_{\mathrm{MN-oAR}}$ to which the attacker has no access). Therefore, although knowing *SK* the attacker is not able to obtain *nMK*. The only other occasion *nMK* is transmitted is from the old AR to the new AR and that communication channel is required to be secured (encrypted and authenticated). Furthermore, as both the old AR and the new AR are not compromised, they will not communicate *nMK* over any other channel. The only remaining information sources are $HMAC_{nMK}(Publ_{\mathrm{A}})$ and $HMAC_{nMK}(Publ_{\mathrm{nAR}})$ contained in the respective *ReAuthTickets*. But as the HMAC function is defined to be a one-way function, both values do not reveal any information about *nMK*. □

As signalling messages are authenticated with key material derived from *nMK*, an attacker is unable to send messages in the name of other nodes. The replay of overheard messages is prevented by the introduced sequence numbers.

As ARs only accept tickets from trusted partners, it is impossible for an attacker to deposit a forged ticket at an AR and authenticate itself based on that ticket. Furthermore, in order to create a re-authentication ticket to a ticket not originated by itself, an attacker would have to know the *nMK* stored in that ticket, which is according to Lemma 1 impossible. For the same reason, it is impossible for an attacker to pose as the AP/AR a MN is supposed to connect to. In order to be able to send messages that are accepted by the MN, the attacker would have to have the ability to produce messages with authentication credentials based on *nMK*.

A last security requirement is that neither the new AR can determine the old key material $MK_{\mathrm{MN-oAR}}$ nor can the old AR determine the new key material $MK_{\mathrm{MN-nAR}}$. As the new AR is not at all involved in the key establishment between the MN and the old AR, it is at most as powerful as a conventional MN and therefore not able to gather any information about $MK_{\mathrm{MN-oAR}}$. The old AR is able to extract $(MN\_ID, nMK)_{SK}$ out of *AuthTicket*. All it needs is *SK* to decrypt *nMK*. But *SK* is only transmitted in *ReAuthTicket* and is encrypted by $Publ_{\mathrm{nAR}}$. As the old AR does neither launch active attacks nor collude with other mobile nodes, *SK* is not revealed to it.

5.2 Performance Considerations

The fast re-authentication protocol consists of only three messages exchanged between the MN and the authenticator (AP or AR). In the IEEE 802.11i scenario two additional messages between the new AP and the new AR accrue. Unlike during the startup with an AAA server involved, the whole communication is held local as the AR generally belongs to the same layer 2 network as the AP.

The computational overhead to compile the fast re-authentication messages is also low. In order to create *AuthTicket*, the mobile node has to draw two random numbers and to perform two symmetric encryption operations. The first message of the re-authentication procedure sent by the new AR is independent from the client and requires no computational resources. A more complex task is the creation of the *ReAuthTicket* as the MN has to use an asymmetric encryption algorithm to encrypt the secret key *SK* with the new AR's public key $Publ_{\mathrm{nAR}}$. The computational overhead can be limited e.g. by using encryption friendly public keys like small RSA exponents. In case the MN already knows $Publ_{\mathrm{nAR}}$ in advance, the re-authentication ticket can also be prepared in advance. The MN may have re-authenticated with that particular AR before and have cached its public key, or the old AR may provide a service that mobile nodes can use to request public keys of certain ARs before initiating HOs. The other operations necessary to compile the *ReAuthTicket* like HMAC-computation are fast operations. With the decryption of *SK* the new AR has to apply only one asymmetric operation. The ticket's decryption and verification require only one symmetric operation and a HMAC computation.

6 Conclusion

In this paper we have addressed the issue of providing a fast mechanism for the re-authentication of mobile nodes performing handovers across access network boundaries. A ticket-based predictive mechanism has been defined that modifies well-established protocol stacks only slightly. In essence, a ticket generated by the MN is forwarded by the current L3 PoA to the destination L3 PoA and the MN re-authenticates itself by proving knowledge of the ticket's contents. The (re-)authentication with an access network is initiated by a modified EAP-Identity-Request called EAP-eIdentity which can be answered by a conventional EAP-Identity-Response or with an EAP-Identity-Response that contains a re-authentication ticket. In the first case, the normal authentication procedure is run whereas in the latter case the MN is immediately authenticated.

Furthermore, we have shown the application of our scheme to IEEE 802.11i like environments and how RADIUS can be used to distribute key material derived from the authentication process. The scheme has been evaluated with respect to security and performance properties. We have shown that our scheme is secure when dealing with single attackers located as mobile nodes in the area covered by the access network. We have further shown that the scheme is efficient in terms of the number of necessary round trips, the time complexity of the underlying operations, and the computational overhead for mobile nodes.

Future work will include hardening the scheme against compromised core network entities and against collusions of mobile nodes with infrastructural devices.

References

1. R. Droms, *Dynamic host configuration protocol*, IETF RFC 1541, March 1997.
2. P. Jayaraman, R. Lopez, Y. Ohba, M. Parthasarathy, and A. Yegin, *PANA Framework*, IETF, Internet-Draft, July 2005. [Online]. Available: www.ietf.org/internet-drafts/draft-ietf-pana-framework-05.txt
3. L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R. Wheeler, *A Method for Transmitting PPP Over Ethernet (PPPoE)*, IETF RFC 2516, February 1999.
4. B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, *Extensible Authentication Protocol (EAP)*, IETF RFC 3748, June 2004.
5. C. Rigney, S. Willens, A. Rubens, and W. Simpson, *Remote Authentication Dial In User Service (RADIUS)*, IETF RFC 2865, June 2000.
6. T. Dierks and C. Allen, *The TLS Protocol version 1.0*, IETF RFC 2246, January 1999.
7. B. Aboba and D. Simon, *PPP EAP TLS Authentication Protocol*, IETF RFC 2716, October 1999.
8. B. Aboba and P. Calhoun, *RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)*, IETF RFC 3579, September 2003.
9. G. Zorn, *Microsoft Vendor-specific RADIUS Attributes*, IETF RFC 2548, March 1999.

Handover Operation in Mobile IP-over-MPLS Networks

Vasos Vassiliou

Department of Computer Science, University of Cyprus,
75 Kallipoleos Str, 1678 Nicosia, Cyprus
vasosv@cs.ucy.ac.cy

Abstract. This paper examines mobility operations, and particularly handovers in an environment, where Hierarchical Mobile IP is operating over MPLS. The work in this paper improves on existing methods of MIP-MPLS interworking first by defining a simple framework based on Hierarchical Mobile IPv6 (HMIPv6) and second by outlining the relevant protocol design assumptions. In addition, the combined protocol is examined in detail under two scenarios and a comprehensive explanation of the operation of intra- and inter-cell handovers is presented.

1 Introduction

The evolution in mobile networks has given rise to several different yet complementary access networks such as second and third generation wireless cellular (2G/ 3G), wireless local area networks (WLAN), and high altitude and satellite networks that offer a broad range of services targeted towards diverse subscriber needs. IP-based wireless networks are a research area of importance since the networks proposed for Universal Mobile Telecommunications System (UMTS) and the next generation (4G) of wireless networks are all-IP based.

To provide satisfactory services to the customers, handover delays, control messages and radio link inefficiencies need to be reduced. Current cell technologies cannot efficiently and cheaply support the density of infrastructure. Innovative interfaces and smaller cells are the solutions to these problems. Smaller cells mean increased signaling when legacy mobility protocols are used. However, in IP-based networks micro-mobility can easily be handled by hierarchical mobile IP (HMIP). The increased requirements of an IP-based radio access network (RAN) can be solved when the scalability and reduced latency of HMIP is combined with the switching performance and traffic engineering capabilities of multiprotocol label switching (MPLS).

The distinguishing feature of MPLS is the ability it offers to users to specify, and tightly control, the communication paths based not only on hop information but also a wide range of Quality of Service (QoS) parameters and policies. Given the tremendous increase in the use of wireless devices to access the Internet and multimedia services, concerns related to providing and maintaining specific service levels arise. It is therefore reasonable to consider an extension of MPLS into the mobile domain.

The goal of the research presented in this paper is to explain how mobility can be introduced, and especially how handovers can be handled, into a Micromobility-enable MPLS network using hierarchical mobile IPv6 (HMIPv6). We consider

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 568 – 579, 2006.

combining the two protocols in an overlay fashion for reasons of simplicity. This work builds on our previous work [5][6] which proposed and examined a framework for the integration of MPLS and HMIPv6 for use in a Radio Access Network.

1.1 Background

*Multiprotocol Label Switchin*g: MPLS [1] is a packet forwarding technology that assigns packet flows to label switched paths (LSPs). Packets are classified at the network edge based on forwarding equivalence classes (FECs). FECs summarize essential information about the packet such as destination, precedence, VPN membership, QoS information, and the route of the packet chosen by traffic engineering (TE). Based on the FEC, packets are labeled, and then transported over a label switched path based on that label. Packets belonging to the same FEC get similar treatment by all intermediate nodes in the path.

MPLS operates between layer two (data link) and layer three (network) of the protocol stack, thus it is referred to as a 2.5 layer architecture. To forward an unlabeled packet MPLS first relates the FEC with an entry in its next hop forwarding equivalence class table (NHLFE). This is done in the FEC-to-NHLFE (FTN) table. The NHLFE table contains the next hop, the operation to be performed on the packet (pop, push, swap) and a new label (if necessary). In a practical implementation the NHLFE also includes the incoming label of a packet so that it can handle labeled packets as well. The resulting table is called the label forwarding information base (LFIB)[1].

Mobile IP: Mobile IP (MIP) allows a mobile node (MN) to move from one link to another without changing the mobile node's home IP address [2]. A home address is an IP address assigned to the mobile node within its home subnet prefix on its home link. Packets may be routed to the mobile node using this address regardless of the mobile node's current point of attachment to the Internet, and the mobile node may continue to communicate with other nodes (stationary or mobile) after moving to a new link. While a mobile node is attached to some foreign network, it is also addressable by one or more care-of addresses (CoA). When away from home, a mobile node registers one of its care-of addresses with a router on its home link; requesting this router to function as the home agent (HA) for the mobile node. The HA intercepts, encapsulates, and forwards packets to the mobile node through its registered CoA.

Hierarchical Mobile IP: Hierarchical Mobile IP (HMIP) is a micro-mobility management model. Its purpose is to reduce the amount of signaling to correspondent nodes and the home agent and improve the handoff speed performance of mobile IP. HMIPv6 [3] is based on MIPv6 [4] and introduces a new entity called the mobility anchor point (MAP), and minor extensions to the mobile node and home agent operations. The major idea is that the mobile node registers the MAP's CoA with its home agent. Therefore, when the mobile node moves locally (i.e. its MAP does not change), it only needs to register its new location with its MAP. Nothing needs to be communicated with the home agent or any other correspondent nodes (CN) outside

[1] NHLFE and LFIB are used interchangeably in this paper.

the RAN. By using this method, signaling is contained in a smaller area, does not overwhelm the core network and the time to complete the location update is smaller.

The rest of the paper is organized a follows. Section 2 presents the design issues that need to be considered during the development of a framework for the interaction of MPLS and hierarchical mobile IP. Section 3 presents our proposed overlay HMIPv6-MPLS framework. Section 4 details the operation of intra- and inter-cell handovers in the resultant framework. Special attention is given to the data delivery processes. Section 5 summarizes the contributions of this work.

2 Framework Design Considerations

2.1 Overview

We distinguish between an overlay and an integrated framework of combining the operations of MPLS and HMIP at the level of interaction between the different architectures. In the overlay method, HMIP, and MPLS remain as separated as possible, without having any merged processes or signaling. Simple events or processes may then require additional messages or additional interaction between architectures to achieve the same result. Therefore, overlay frameworks usually introduce more latency and overhead. On the other hand, an integrated framework merges and relates many of the functions of its composing members. There exists a tradeoff between the simplicity of operation and the optimization and system performance. In this work we are focusing on the overlay paradigm. Our solution for the integrated paradigm can be found in [5] and [6].

2.2 Radio Access Network

The basic topology for this research work is a Radio Access Network (RAN) as shown in Fig. 1.

The RAN consists of at least three layers of label switched routers (LSRs). The edge components of the architecture are the radio access routers (RAS), which are the first IP aware devices of the network seen from the mobile terminal. One, or more, base stations (BS) are attached to a RAS (or integrated into it) and provide the physical radio link to the mobile node (MN). Several RASs are interconnected to one or more Edge Gateways, which in turn provide access to outer (backbone) networks including other RANs. The RASs and the EGWs are linked through a network of MPLS-capable routers. We assume that all routers in the RAN can act as mobility agents (MA) to support mobility management based on hierarchical mobile IP.

The design of the reference network has been based on network architectures used by well-established providers and was also influenced by traffic engineering considerations as these are described in [7]. Enhancements to the network may include additional layers of hierarchy and more complex interconnection like full mesh or double homing.

Fig. 1 shows a general multi-RAN scenario where the mobile node traverses the center RAN. References will be made to nodes in this schematic in later sections.

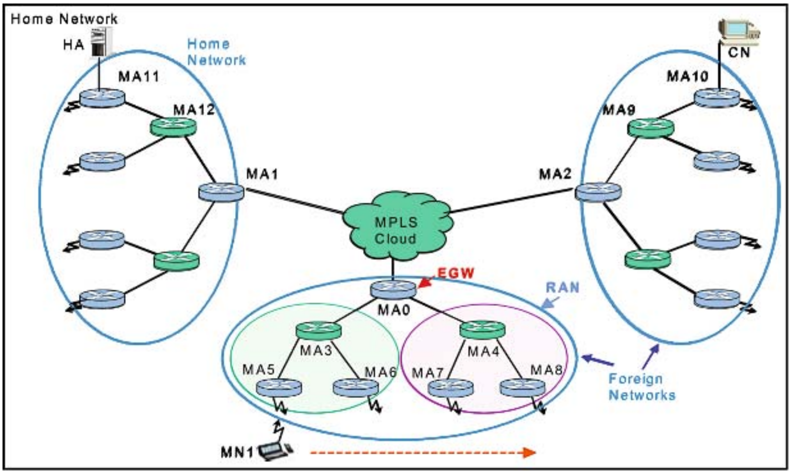


Fig. 1. Reference Network

2.3 Design Assumptions and Specifications

The design of the MIPv6 and HMIPv6 based MPLS framework is based on the following assumptions:

- All MPLS nodes in the RAN are mobility-enabled
- Mobile IP procedures for agent discovery, mobile node registration, and routing remain unchanged.
- Mobile nodes have no MPLS related protocols in their stack
- Only point-to-point LSPs are considered.
- MPLS operates in the following modes:
  - Downstream on demand: An LSR explicitly requests a label binding for an FEC from its next hop for that particular FEC.
  - Ordered control: An LSR only binds a label to a particular FEC if it is the egress for that FEC, or if it has already received a label binding for that FEC.
  - Conservative retention: An LSR discards any label bindings from downstream routers if those routers are not its next hop (or no longer its next hop) for a particular FEC. This retention mode requires an LSR to maintain fewer labels.
- There is a unique label per LSP (i.e., there is no label merging). An LSR can support label merging if it has bound multiple incoming packets to an FEC that uses a single outgoing label. Once packets are transmitted using this method there is no way of differentiating them in terms of their source (input interface or incoming labels). Without label merging, if two packets for the same FEC arrive with different incoming labels they must be forwarded with different outgoing labels.
- No aggregation allowed (i.e., more than one LSPs for the same FEC is acceptable). Aggregation is the procedure of binding a single label to a union of FECs; it is itself an FEC (within a domain).

- In our framework we would like to be able to have the finest granularity of label switched paths. For that reason we allow more than one LSPs for the same FEC from the same end node. FECs are defined on end-node pairs and QoS requirements.
- No penultimate hop popping is considered.
- The Data-driven method is used for the establishment of paths in a mobile network. An LSP is established only if data needs to be transferred between nodes.

We assume that the mobile node has already done address auto-configuration and registration, and has received an acknowledgement from its home agent. The CoA registered with the home agent is that of the mobility agent located at the edge gateway. Before we get into the details of the framework we have to consider other design issues.

Binding Update – LSP Setup Relationship

The relationship between the home agent binding update procedure and the home agent-mobility agent LSP setup can take many forms. Previous work has considered two methods for combining them: *sequential* (Fig. 2a) and *encapsulated* (Fig. 2b). In the *sequential* method the two procedures are initiated one after the other, with the LSP setup following a successful binding update (exchange of BU and BUAck). In the *encapsulated* method the home agent initiates the LSP setup after it receives the binding update message from the mobile node. However, a binding update acknowledgement message is not sent back until the LSP setup is complete.

- The sequential method of registering the MN to HA and establishing the related LSP is used.[2]

Binding Update Acknowledgement

Most of the current proposals use binding acknowledgement messages during Registration, even though they are not a requirement of the mobility protocol.

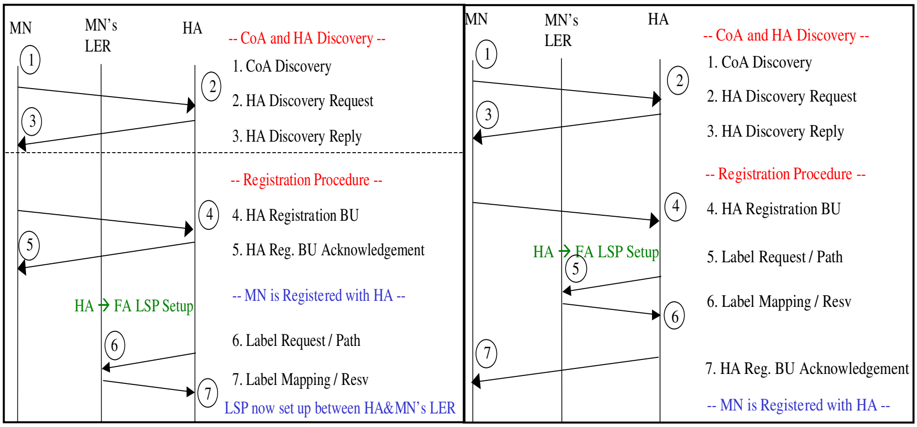


Fig. 2. HA-MA LSP setup. (a) Sequential Method and (b) Encapsulated Method.

[2] For a more comprehensive justification of the design points, see [9].

We believe that LSPs should be set up after a binding update acknowledgement is sent back from the correspondent node, so as to avoid setting up LSPs for connections that are not going to be accepted.

Not setting the LSP before an acknowledgement is also helpful in an Integrated Services QoS environment since the other end of the connection along with the intermediate routers needs to be consulted. If we allow the mobile node to move into neighboring domains, we may want to utilize this option to make sure that any LSPs to the mobile node are set up only after the service level agreements between the different domains and the mobile node are fully negotiated and accepted. Even though we do not address them in this work, security considerations are also a factor for expecting an acknowledgement from the home agent or other nodes before proceeding.

- LSPs are set up after a binding update acknowledgement is sent back from the correspondent node.

3 Hierarchical Mobile IP - MPLS Overlay Framework

The overlay framework we propose is based on [3], [6], [8] and the assumptions and requirements stated in Section 2.3. In terms of architecture, we consider co-locating the mobility agents with the LSRs in the RAN. The interaction of the two is limited to the LSRs using the routing tables updated by HMIP. Procedures like HMIP registration and LSP setup are independent and databases do not share entries or reference each other.

The following subsections explain the protocol based on the specifications descrybed above. We consider two slightly different scenarios: one in which (some) IP packets are allowed to traverse the network and one in which only MPLS is used as the transport.

3.1 Scenario 1 – MPLS Data Packets Only

The basic signaling diagram for the scenario where only MPLS data packets are allowed is shown in Fig. 3.

3.2 Scenario 2 - IP Data Packets Allowed

From a closer examination of the first case in scenario 1 (Fig. 3), it seems that a lot of messages are communicated to set up a number of initial LSPs between the correspondent node, the home agent, the MAP and the mobile node's LER. All these LSPs are not used for long since the correspondent node creates a more direct path to the mobile node soon after it receives a binding update. Therefore, we consider the scenario where some IP packets are allowed to traverse the MPLS network. The information allowed to be communicated in pure IP format is limited to the packets originating from a correspondent node and directed to the home address of a mobile node when there has not been an LSP set up for such communication before.

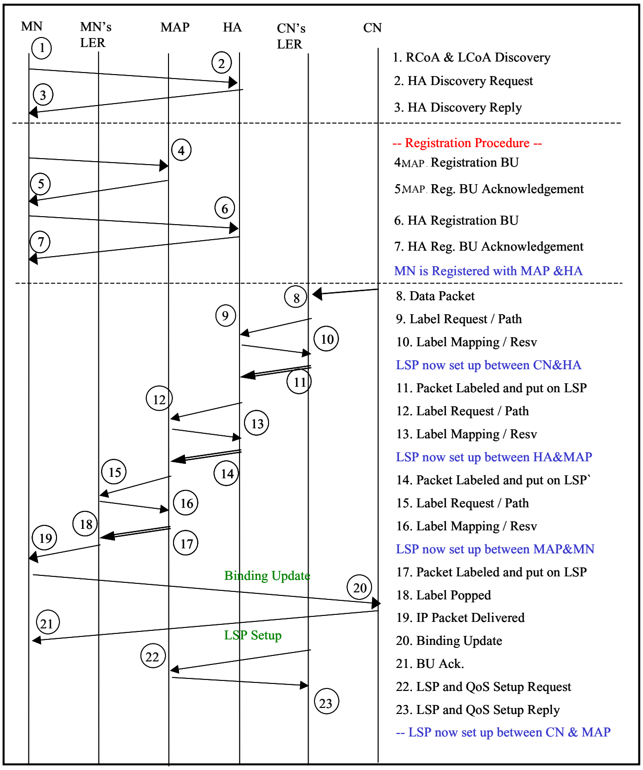

MN
MN's LER
MAP
HA
CN's LER
CN
1. RCoA & LCoA Discovery
2. HA Discovery Request
3. HA Discovery Reply
-- Registration Procedure --
4MAP Registration BU
5MAP Reg. BU Acknowledgement
6. HA Registration BU
7. HA Reg. BU Acknowledgement
MN is Registered with MAP &HA
8. Data Packet
9. Label Request / Path
10. Label Mapping / Resv
LSP now set up between CN&HA
11. Packet Labeled and put on LSP
12. Label Request / Path
13. Label Mapping / Resv
LSP now set up between HA&MAP
14. Packet Labeled and put on LSP`
15. Label Request / Path
16. Label Mapping / Resv
LSP now set up between MAP&MN
17. Packet Labeled and put on LSP
18. Label Popped
19. IP Packet Delivered
20. Binding Update
21. BU Ack.
22. LSP and QoS Setup Request
23. LSP and QoS Setup Reply
-- LSP now set up between CN & MAP
Binding Update
LSP Setup

Fig. 3. HMIPv6; Data Driven HA-MA LSP; MPLS data only

3.2.1 Correspondent Node Initiates Transmission

There are three slightly different cases considered for this scenario:

- Case 1. CN has no binding for the MN, and CN's LER has no LSP to MN's address
- Case 2. CN has no binding for the MN, and CN's LER has LSP to MN's address
- Case 3. CN has a binding for the MN, and CN's LER has LSP to MN's new location

When a data packet arrives, or is created at the correspondent node, the node first examines its binding cache for an entry of mobile node's home address. If the correspondent node does not have an entry it sends the packet to the mobile node's home address. If MA10 does not have an LSP already established between the correspondent node and the mobile node (case 1), it proceeds by sending it using pure IP as described in the basic HMIP operation. The Regional CoA associated with the mobile node is that of MA0. When the mobile node receives a tunneled message it will send a binding update for its Regional CoA to the correspondent node, which will use it for the establishment of an LSP to the mobile node if there is a need.

This operation reduces the amount of MPLS related overhead at the initial stages of a communication. At the same time, no MPLS based QoS support is provided to those packets. DiffServ support in IP using the DSCP fields in the IP header could be used in such circumstances. In addition, if the CN-MN LSP is set up quickly, then there may be no reason for concern even if no QoS is provisioned for those packets.

If MA10 does have an LSP set up toward the mobile node's home address (case2), it uses MPLS to send the packet. The home agent will intercept the packet after the MPLS header is stripped off and recognize that there is an entry in its binding cache for that mobile node. It will then create an encapsulated header and send the packet to the mobile node through the MAP. The rest of the operations are as in the first case.

CN's LER – MA10

Input I/F	Input Label	FEC	Operation	Out I/F	Out Label
--	--	MN1	Push	1	10

MN's HA – MA11

Input I/F	Input Label	FEC	Operation	Out I/F	Out Label
1	10	MN1	Pop	--	--

In the third case, the correspondent node uses the mobile node's Regional CoA from its binding cache. The CN's LER (MA10) will find the corresponding entry in the LFIB table and form MPLS packets with label 21 as the outgoing label.

CN's LER – MA10

Input I/F	Input Label	FEC	Operation	Out I/F	Out Label
--	--	MN1	Push	1	10
--	--	RCoA	Push	1	21

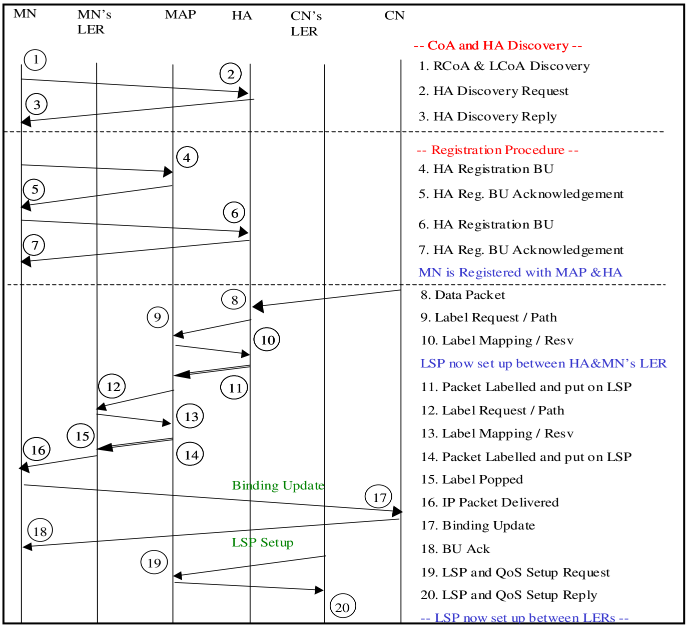


Fig. 4. HMIPv6; Data Driven HA-MA LSP establishment; IP Packets allowed

3.2.2 Mobile Node Initiates Communication

In this scenario (shown in Fig. 4), some IP packets can be routed from the mobile node to the correspondent node without using MPLS. The mobile node's LER (MA5) will forward pure IP packets until it recognizes that a flow is in place.

4 Handovers in the Overlay Framework

When a mobile node moves out of the range of a mobility agent and into the range of another, the movement is understood by the difference in the router advertisements received. If the movement is inside the RAN (below the MAP), the handoff is an intra-RAN handoff. If the change in position is such that a new MAP is going to be used, the handoff is called inter-RAN handoff.

Regardless of the type of handoff (intra- or inter-RAN), there are many ways to perform path rerouting following a location change:

- LSP re-establishment – A new LSP created to/from the new location.
  - Advantages: simple
  - Disadvantages: Packets in transit are lost
- LSP extension – The LSP is extended from the old edge router to the new edge router
  - Advantage: fast, no packets are lost
  - Disadvantages: LSP length is increased, delay may be increased, loop prevention is required
- LSP extension and modification – the LSP is first extended and then modified.
  - Combination of the first two methods
  - Advantages: Fast, simple, no packets are lost
- LSP multicast – LSPs are created in multiple locations around the MN
  - Advantages: enables fast and smooth handoff
  - Disadvantages: point-to-multipoint LSPs needed, extensive location knowledge is required. Difficult to handle resources
- LSP dynamic rerouting – the LSP is modified starting from the lowest (closest to the MN) common router between the old and the new path. Partial path re-establishment.
  - Improvement on extension and modification method.
  - Advantages: full LSP re-establishment is not required. Parts of LSP remain the same, which means less signaling.
  - Disadvantages: increased complexity

All of the rerouting methods described above have been used in current proposals. Multicasting and dynamic rerouting seem the most popular and efficient. However, the corresponding proposals make certain assumptions on the capabilities of LDP or the level of interaction between architectures that do not fit the realities of our model of an overlay framework. Therefore, we propose the use of the LSP extension and modification method. This method is simple; it can be used without any additions to the architectures; uses existing functions (like the old MA notification and MN-CN BU) as the basis; and also limits packet loss.

4.1 Intra-RAN Handoff

Suppose that MN1 moves out of the range of MA5 to a location closer to MA6. The mobile node will obtain a new LCoA from its new location. It will then send a binding update to MA5 with the new LCoA so that packets in transit toward the MA5 are redirected toward MA6. At the same time, the mobile node will send a local binding update to its MAP (MA0) and any correspondent nodes inside the RAN. Prior to the handoff, MA5 has the following data in its LFIB:

Old LER - MA5 (before handoff)

Input I/F	Input Label	FEC	Operation	Out I/F	Out Label
1	30	LCoA1	Pop	--	--
1	31	LCoA1	Pop	--	--
--	--	CN	Push	1	40

After the handoff MA5 will initiate an LSP setup between itself and MA6. The LSP will be for LCoA2. MA5 and MA6 will update their tables accordingly.

Old LER - MA5 (after handoff)

Input I/F	Input Label	FEC	Operation	Out I/F	Out Label
1	30	LCoA1	Pop	--	--
1	31	LCoA1	Pop	--	--
--	--	CN	Push	1	40
--	--	LCoA2	Push	1	50

New LER – MA6

Input I/F	Input Label	FEC	Operation	Out I/F	Out Label
1	50	LCoA2	Pop	--	--

The result of this extension is the aggregation of the MN LSPs at MA5 into one LSP for MA6. We do not consider this to be an important issue given that the path is only going to be used for the limited number of packets in transit (or for packets originating from MA5).

Prior to an intra-RAN handoff the MAP has an entry in its binding cache relating the RCoA and LCoA used by the mobile node, and an LSP connecting to the LER serving it. After the handoff the MAP has a different LCoA associated with the mobile node and needs to establish a path toward it. If full LSP re-establishment is used, the MAP will establish a new LSP toward the mobile node and add the entry in the LFIB. The connection between the old and the new entries is done in the binding cache. Therefore, the trend of leaving the MPLS layer in order to get the new CoA and return to find the new path to it is continued here. As with the old LER, the operation will end up aggregating all of MAP's existing LSPs to the mobile node into one, unless the MAP performs an LSP setup for every entry it has in its table. In this case, to differentiate between entries the FEC needs to contain the address of the sender node as well. Therefore in the overlay environment we also add the requirement that FECs denote end node pairs.

4.2 Inter-RAN handoff

Inter-RAN handoffs include everything done in intra-RAN handoffs with the addition that the mobile node's home agent and correspondent nodes outside the RAN will have to establish LSP(s) to the mobile node's new MAP. Let us consider the case where MN1 moves into the RAN served by MA2. The home agent will receive a binding update with MA2 as the new RCoA and update its binding cache with the new value. If its connection with the mobile node is active (data present) it will also initiate an LSP setup to the new RCoA. Correspondent nodes outside the mobile node's new RAN will also have to do the same. The label forwarding information base of the HA will be changed to:

HA – MA11

Input I/F	Input Label	FEC	Operation	Out I/F	Out Label
1	10	MN1	Pop	--	--
--	--	RCoA1	Push	1	20
--	--	RCoA2	Push	1	60

The entry for the old RCoA will remain in the table until released by the ingress or withdrawn by the egress router. There is no provision at present for the release or withdrawal of these labels based on mobile IP information.

Since MA0 will always be the correct downstream router for RCoA1 MPLS does not give the option to the upstream router to release the label. A downstream node can withdraw a label if it decides to break the binding between the label and the address prefix associated with it. The LSR withdrawing a label must do so from every LSR it has distributed that label. Label withdrawing is useful in the handoff framework only if there is a mechanism to inform MPLS that the binding is not needed anymore. This is another example where triggering MPLS events through mobile IP events is beneficial.

5 Conclusions

This paper proposed a framework that integrates Multi-Protocol Label Switching (MPLS) and Hierarchical Mobile IPv6 (HMIPv6) in a Radio Access Network (RAN) in a simple overlay fashion. The need for such a framework stems from the increased drive toward high-speed multimedia-intensive services.

The overlay method proposed improves on existing methods of MIP-MPLS interworking based on MIPv4 and HMIPv4 and utilizes HMIPv6 as the micromobility protocol.

Detailed operation signaling diagrams as well as forwarding table contents were presented and the ability of the protocol to handle mobility events (handover) was illustrated both for the intra and inter-RAN cases.

In conclusion, we find that MPLS, when paired with a suitable mobility protocol can function well in a radio access network and provide the same benefits it offers when used in wired networks.

Acknowledgements

This work has been partly supported by the European Union under the project E-NEXT FP6-506869.

References

[1] E. Rosen, A. Viswanathan and R. Callon, "Multiprotocol Label Switching Architecture," Request for Comment 3031, Internet Engineering Task Force, Jan. 2001.

[2] C. Perkins Ed., "IP Mobility Support", Request for Comment 3220, Internet Engineering Task Force, Jan. 2002.

[3] H. Soliman, C. Castelluccia, K. El-Makri, and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," Request for Comment 4140, Internet Engineering Task Force, Aug. 2005.

[4] C. Perkins, D. Johnson, and J. Arkko "Mobility Support in IPv6," Request for Comment 3775, Internet Engineering Task Force, Jun. 2004.

[5] V. Vassiliou, H. L. Owen, D. A. Barlow, J Grimminger, H-P Huth, and J. Sokol, "A Radio Access Network for Next Generation Wireless Networks Based on MPLS and Hierarchical Mobile IP," In Proc. IEEE 56th Vehicular Technology Conference Fall 2002 (VTC2002-Fall), Vancouver, Canada,. pp 782-786, Sep. 2002.

[6] V. Vassiliou, H. L. Owen, D. A. Barlow, J Grimminger, H-P Huth, and J. Sokol, "M-MPLS: Micromobility-enabled Multiprotocol Label Switching," In Proc. IEEE International Conference on Communications (ICC2003), Alaska, USA, May 2003.

[7] Barlow, D., Vassiliou, V., Krasser, S., Owen, H., Grimminger, J., Huth, H.-P., Sokol, J., "Traffic Engineering Based on Local States In Internet Protocol-based Radio Access Networks", Journal Computer Networks, Vol. 7, No. 3, September 2005, pp. 377-384.

[8] J. K. Choi, M.H. Kim, and T.W. Um, "Mobile IPv6 support in MPLS," Internet Draft <draft-choi-mobileip-ipv6-mpls-01.txt>, Aug. 2001.

[9] V. Vassiliou, "An Integration Framework and a Signaling Protocol for MPLS/ DiffServ/ HMIP Radio Access Networks," Ph.D. Thesis, Georgia Institute of Technology, Jul. 2002.

The Design and Implementation of a Quality-Based Handover Trigger

Ian Marsh[1], Björn Grönvall[1], and Florian Hammer[2]

[1] SICS, Kista, Sweden
{ianm, bg}@sics.se
[2] Telecommunications Research Center (ftw.) Vienna, Austria
hammer@ftw.at

Abstract. Wireless connectivity is needed to bring IP-based telephony into serious competition with the existing cellular infrastructure. However it is well known that voice quality problems can occur when used with unlicensed spectrum technologies such as the popular IEEE 802.11 standards. The cellular infrastructure could provide alternative network access should users roam out of 802.11 coverage or if heavy traffic loads are encountered in the 802.11 cell. Therefore, our goal is to design a handover mechanism to switch ongoing calls to the cellular network when the 802.11 network cannot sustain sufficient call quality. We have investigated load and coverage scenarios and designed, implemented and evaluated the performance of an 802.11 quality-based trigger for the handover of voice calls to the cellular network. We show that our predictive solution addresses the coverage problem and evaluate it within a real setting.

Keywords: VoIP, 802.11-cellular convergence, quality prediction.

1 Introduction

Handsets that are equipped with multiple standard radios will become commonplace. PDAs with 2G cellular radios and IEEE 802.11 chipsets are already on the market, and dual-radio mobile phones are also beginning to appear. The primary motivations for a voice handover system are monetary. By connecting to 802.11 access points when available, it should be possible to avoid cellular tariffs. However when users leave the 802.11 area they may want to continue their voice calls. Therefore a handover mechanism to alternative technologies for voice users is desirable. Excess traffic within an 802.11 cell is also a reason to handover a call to the cellular system. The basic problem is when to perform, or even schedule, a handover from one system to the other. The cellular infrastructure provides network support for its clients, and performs the handover on their behalf. The clients periodically report their reception status enabling the infrastructure to make an informed handover decision. In an 802.11 system this functionality is not available, therefore it becomes the task of the handset when best to handover a session. Prediction is the key issue with this approach as voice call setup takes approximately five seconds to the fixed or cellular network. This

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 580–591, 2006.

is an average value we observed by repeatedly calling to the PSTN and GSM networks. During the handover, ideally no quality differences should be audible making the handover as transparent as possible. On the other hand, the system should not handover voice calls to the cellular system due to small audio glitches that many mobile users have become accustomed to, worse still switch back or forth between network types. Manual switching should always be an option, if users want to use the cellular network. However, in this work we assume that users want to use the 802.11 networks for voice communication when available. Therefore the contribution of this work is an *automatic* handover solution for real-time voice sessions on 802.11 networks to the cellular infrastructure when poor quality conditions persist.

2 Assessing the Influence of Packet Loss Using PESQ

Packet loss is critical when determining voice quality. Bursty losses are well known to be commonplace in wireless communication, and 802.11 networks are no exception. Therefore the goal of this first evaluation is to ascertain how many packets can be lost in a burst without significant reductions in the perceptual quality. We do not consider delay or jitter in this first phase, only packet losses.

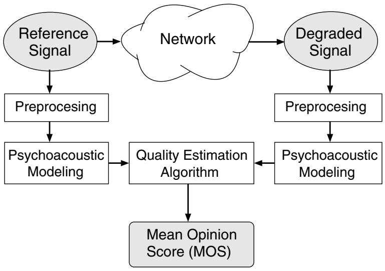


Fig. 1. The PESQ processing structure

PESQ MOS	Linguistic equivalent	Quality degradation
4.5	Excellent	None
4	Good	
3.5	Good/Fair	Moderate
3	Fair	
2.5	Fair/Poor	Severe
2	Poor	
1	Bad	

Fig. 2. A quality degradation scale

Figure 1 shows the functional units of PESQ, the ITU-T standard we derive our loss tolerances from [6]. A reference speech signal is transmitted through a network that results in a quality degradation corresponding to the path conditions and coding scheme. PESQ analyzes both the reference and degraded signal and calculates their representation in the perceptual domain based on a psychoacoustic model of the human auditory system. The disturbance between the original and the degraded speech signal is calculated by the quality estimation algorithm and a corresponding subjective Mean Opinion Score (MOS) is derived. The evaluation of speech quality using PESQ is performed off-line due to its computational complexity. For example a 400 packet sequence with ten losses requires approximately two seconds of processing time for simple G.711 coded speech. G.711 yields the maximum PESQ score (4.5) in the absence of loss, however it is particularly sensitive to packet loss even with concealment. We have evaluated the tolerable loss lengths using both G.729 and iLBC, but they were always less than G.711, i.e. G.711 can be considered a worst-case codec. It is also the format used in our fully integrated solution, and thus allows us to directly set the loss thresholds in the handover trigger function without any transformation.

Figure 2 shows the PESQ MOS scale as defined by the ITU and their English linguistic equivalents. We have added an extra column, quality degradation, to indicate the quality reductions we have looked at as part of this first phase. The degradation of a MOS point is referred to as "moderate" and two points as "severe". We degrade the complete ITU-standardized eight second speech sample with 1 to 50 continuous losses. For each of the 50 loss bursts, we record the MOS score, and then shift the pattern through the eight second sample until it has been completely assessed for loss sensitivity. The technique and its effectiveness is fully described in [2]. Since the results are highly influenced by the performance of the packet loss concealment (PLC) algorithms, we conducted the tests with and without PLC. The loss concealment algorithm used was the one standardized by the ITU for G.711 called G.711i [5].

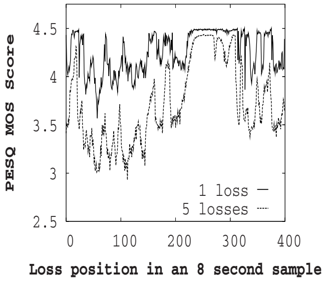


Fig. 3. Singular & quintuple loss scores for a female English speech sample

Quality reduction	Gender	Language		
		English	French	Japanese
Moderate	Male	3/7	9/12	4/8
	Female	4/7	4/8	3/8
Severe	Male	30/31	43/45	45/46
	Female	31/32	46/48	45/48

Fig. 4. Packet loss lengths for 1 & 2 MOS reductions. The first value of the X/Y pair is without using PLC, the second is with PLC.

Examples of single and quintuple consecutive loss lengths with loss concealment are shown in Figure 3. The sample is one from the ITU standard database and is an American English female, the text is "She broke her new shoelace that day, the coffee stand is too high for the couch" and lasts for seven seconds. Observe that the concealment works well for one lost packet, however five consecutive losses are more difficult to conceal hence resulting in a lower PESQ score. Also note the silence period between samples 225 and 300 corresponding to the pause between the two phrases. The results for three different languages are given in Figure 4. The 90% percentile was taken for the MOS scores. As one can see the maximum number of consecutive packets one should allow in a burst without PLC is three for a moderate drop in quality for an English female speaker. However in reality loss concealment is employed in the receiver, in our full working system too, so we take seven as the threshold. It can be seen that English is the most sensitive amongst these three particular samples.

3 Emulating a Mobile System

We move onto understanding the effect of other parameters on the design of a handover trigger by creating an experimental testbed. Our experiments have three major goals, first to gauge the impact of distance on wireless VoIP communication, second to understand the dynamics of voice streams mixed with TCP downstream traffic, and third how to measure and combine the available metrics suitable for implementing a handover trigger. Figure 5 shows the setup, it consists of a mobile terminal, a server we call a PBX, and load generating nodes. The PBX connects VoIP calls to the Public Switched Telephone Network (PSTN) and has the capability to handover calls to the cellular network when requested. The PBX and load generator are on a 100 Mbits/sec Ethernet, the mobile node and the sink are on the 802.11b network.

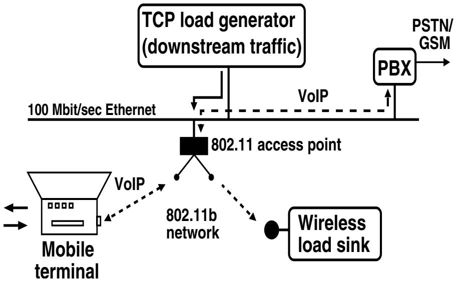


Fig. 5. The experimental testbed setup used in emulating a system capable of handover

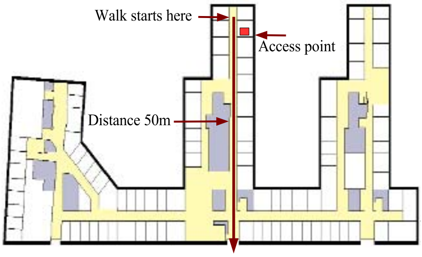


Fig. 6. The office layout used for our quality tests with "walk" marked

The target network is expected to be used for voice applications, but also for traditional TCP-based applications such as email and web surfing. Therefore, we have developed a TCP NewReno load generator which attempts to create flows targeting a specified rate when network resources permit. For our stated goal of the design of a quality-based handover trigger, we will now explain three separate experiments:

Fading signal experiment: In this setup the mobile terminal moves past an access point and out of its coverage area. This is shown in Figure 6 as the arrowed line. The mobile terminal was carried along a corridor at walking speed and away from the access point. The left and center plots within Figure 7 show how voice packets arrive late or are lost due to environmental variations.

From the figure we can see that during normal interference conditions there is little packet loss. As the signal deteriorates however, losses become much more frequent and the length of the loss bursts increase. One interesting observation is that packet losses occur much earlier at the mobile than at the PBX, compare the left and center plots. We assume this is due to better reception capabilities at the access point, for example better gain in the antennas or a dual antenna approach provides more diversity for receiving weak signals.

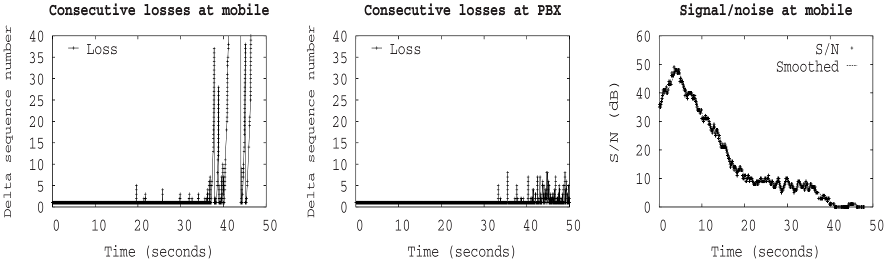


Fig. 7. Consecutive losses observed by the moving terminal (left) and the PBX (center) and signal strength as reported by the terminal (right)

Packet losses are first experienced at the mobile, but in the target system it is the PBX that will perform the handover. This is because the functionality to handle both PSTN and IP calls is within the PBX. Therefore the PBX needs to be *continuously* monitoring the signal and network conditions at the mobile. This information can be sent either by piggybacking data onto the voice packets, or by sending RTCP-like designated packets at fixed time intervals as we do. From a system design perspective, it is critical that the PBX knows the state of the mobile.

The right plot of Figure 8 depicts how the mobile varies the transmission rate over time. During good signal conditions the mobile always uses the maximum transmission rate. During reasonable conditions the mobile varies the rate as it discovers link layer retransmissions become necessary [8, 9]. During poor conditions it constantly transmits at 1 Mbit/s. Notice that 1 Mbit/s is a *critical point*, as at this point it could lose connectivity altogether. Thus, when transmitting at 1 Mbit/s, a handover to the cellular network should be considered imminently, however it is not necessarily true that operating at 1 Mbit/s implies poor quality. Observe that a handover to the cellular network should ideally have completed at $t = 36$, which would have meant scheduling the handover approximately at $t = 31$ (the left plot of Figure 8), otherwise, poor quality could be experienced before the cellular call is in progress.

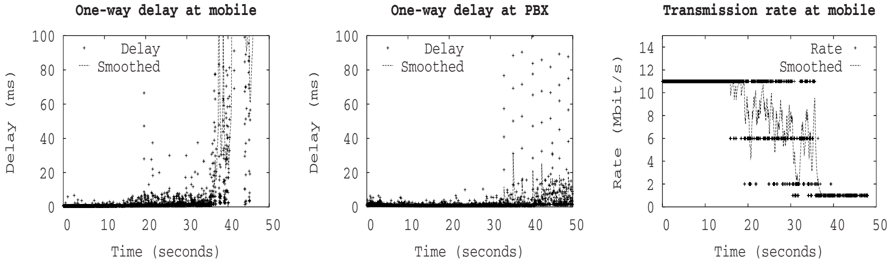


Fig. 8. Delays at the moving terminal (left) and the PBX (center) and changing transmission rates recorded at the terminal (right)

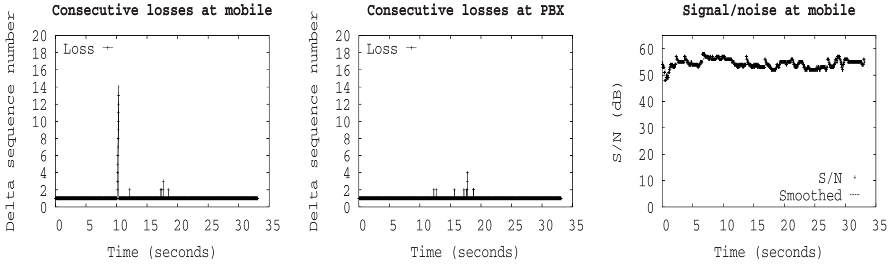


Fig. 9. Losses recorded at a stationary terminal (left) and a stationary PBX (center) shown with signal strengths (right) on a **loaded** network

Loaded network: In this experiment we study the effects of a network operating close, but below, its full capacity. The synthetic load is limited to a target rate by our load generator. Due to the TCP behavior, the network will be overloaded for short periods of time. The synthetic load is directed towards (into) the 802.11 network in order to simulate web browsing or an email download. In this experiment we monitor an ongoing call and after ten seconds add synthetic load so that the network is operating at almost its full capacity. After a further ten seconds stop the synthetic load. In the left plot of Figure 9, we observe how the mobile at time $t = 10$ experiences a contiguous sequence of 13 packets that are delayed for more than 20ms and are effectively lost. This is because we used a constant size jitter buffer of 20ms in both the terminal and PBX.

At the same time, we can see in the left graph of Figure 10 how the delay increases for each packet on its way from the PBX to the mobile, a queue is building up in the access point. The web servers are sending more packets into the 802.11 network than it can handle, and it takes time before TCP reacts and consequently backs off. During this time a queue builds up as packets arrive on the fixed network and they must be enqueued before gaining access to the congested 802.11 network. Since voice packets are delayed behind the TCP packets, they will eventually arrive late at the mobile. From the center graph of Figure 10, we can see how the delay from the mobile terminal towards the PBX increases when the network is loaded. The increase in delay is a result of the 802.11 contention, however in this case there is no extra queuing in the access point as the 100 Mbits/sec Ethernet is much faster than the 11 Mbits/sec 802.11b network. The asymmetry in the network speeds is clearly evident in these two cases. To conclude, we observe that loss events in either direction are rare, even in a loaded network. For these loss events, the burst-loss length is typically one, and these can be dealt with using standard loss concealment methods such as G.711i.

Overloaded network: In a continuation of the previous experiment, but with a synthetic load driving the network to its maximum operating capacity. These figures are not included in the interests of space, but are briefly described. In these experiments, we observe serious loss problems from the PBX to the mobile, but not in the reverse direction, i.e. from the mobile to the PBX. This is to be expected, as we are again observing a queue building up in the access points

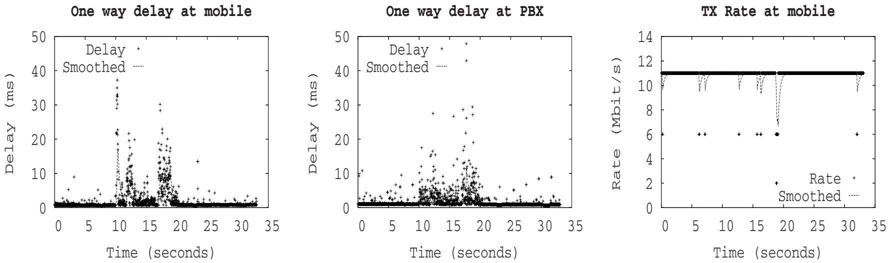


Fig. 10. Delays recorded at a terminal (left) and a stationary PBX (center) shown with transmission rates (right) on a **loaded** network

as TCP packets arrive. It is trivial for the mobile to detect this and promptly inform the PBX since the traffic from the terminal to the access point is still unhindered. The problem arises with the speed this can occur. The network load can increase from unloaded to full capacity in a fraction of a second, however as we know it takes several seconds to establish a call through the PSTN. A better solution than triggering a handover in this case, is to give the voice traffic higher priority e.g. by marking the speech packets as having priority as proposed by the IETF Differentiated Services framework, for example by using the Expedited Forwarding (EF) class of service. The access point must also be capable of detecting these and scheduling the appropriate priorities.

4 Handover Design and Implementation

We now consider our real system with voice-enabled PDA's using commercial software, firmware and hardware solutions. When using real systems, the availability, reliability and resolution of network and link layer metrics are not the same on all systems. Therefore we chose not rely on one or two metrics rather to use a linear combination of those available for our trigger mechanism. Ideally we would like to use as many as possible *and* reliable, but certain hardware and software limitations prohibit this. The advantage of using this kind of combination is if the value is not available or reliable it contributes nothing, i.e. 0 to the overall score. The single value to make the handover decision we refer to as the **handover score**. The usable metrics we call the **handover contributors** and rationalize their inclusion in the following paragraphs. The scores are derived from numerous experimental and empirical investigations as previously described.

Importance of periodic reporting: We have previously stated the terminal should report to the PBX the current quality conditions it is observing. Loss and jitter metrics are sent every 0.5 sec from the mobile terminal to the PBX. Link layer metrics are read at intervals of 0.125 sec, four times the frequency of the VoIP metrics. Since the link layer situation ultimately reflects in the quality seen at the application layer, we deemed it necessary to use higher resolution at this layer. The link layer metrics are averaged and sent with the network parameters in RTCP-like reports. The timings are a tradeoff between the measurement resolution and the CPU load on the PDA.

Signal strength: As we have seen the signal to noise ratio is a good indicator of potential problems. Therefore given a dependable reading, we only need to record its value and scale it to our handover score. Unfortunately the signal strength reading from the PDAs tends to bottom out long before we loose connectivity, and consequently only makes a small contribution to the handover score, which is a limitation of the terminals we used. A positive signal strength is simply added to the score, in our experiments with the HP terminal this varied between +90 and 0.

Loss: We have seen from our off-line PESQ experiments that eight losses are sufficient to reduce the quality from "excellent" to "good/fair". A 20ms packetisation represents 50 packets per second, therefore a loss of eight packets corresponds to a loss percentage of 16% percent. In each second there are two reports (0.5 sec per report), therefore a loss of 8% should be taken into account. A score of -10 is attributed to this degree of loss for each interval and an additional -10 is added if this level spans over two intervals.

Jitter: We have seen increasing jitter was the best indicator we had of poor upcoming quality. In an open system it is easy to calculate the mean and variance of the VoIP stream by observing packet interarrival times. However, in our full system jitter estimates are returned from a commercial VoIP encoding and playout system called the GIPS Engine[1]. We were uncertain about the exact units returned, but found from experimentation that, values between 0-68 signified good conditions, whilst those between 69-80 were interpreted as neutral, 81-93 as bad and over 94 as poor. To find these values we loaded the network as described in the emulated cases, and observed the values reported. We attributed scores of +10 to the good conditions (i.e. a positive score), 0 to the neutral situation, -10 and -20 for the poor and very poor situations respectively. Similarly if these conditions span over two intervals, this is accounted for in the score.

RTCP losses: It is important that the PBX has information about the state of the mobile terminal, as if the PBX is not receiving reports then the mobile is probably having reception problems and as we have seen, more likely worse than those seen at the PBX. Therefore sending regular reports from the mobile terminal to the PBX probes the 802.11 quality, and reports indicate potential problems. We chose three or more consecutive losses as sufficiently significant to reduce the score. Two or more report losses are interpreted as poor conditions between the handset and PBX and a score of -10 is attributed to this condition.

Transmission rates: As the system reduces the rate we would ideally like to reflect this in the handover score. In particular changes to the lower rates i.e. 2 and 1 Mbits/sec should reduce the score as the probability of a connection loss increases. However the PDA terminals did not reliably report this value to our

[1] http://www.globalipsound.com

application, hence we could not include it into our score function. As we have shown, laptops in the testbed setup gave IEEE transmission rates that we could have been used.

Handover score weighting: Since we have chosen to use a linear combination of the metrics, it is simple a matter of combining the above metrics into a single score value.

$$Handover\ score = Signal + Loss + Jitter + Report\ losses$$

Handover score values: For convenience our implementation uses a handover score that varies between -100 and 100. A large positive value indicates good quality. The user enters a threshold value and a handover will occur when the score falls below this level. We chose +30 as a default from experimental testing found it to be satisfactory. By increasing the threshold, average quality will improve but at greater expense since the system will hand over the call to the GSM system earlier. Conversely by decreasing the threshold, GSM expenses will be reduced but the periods of degraded audio quality will be longer. It was necessary to smooth these scores in some cases by considering two intervals, however we attribute this to using some combinations of hardware. This was not necessary in the emulated testbed setup.

5 System Evaluation

In this section we describe the procedure we used to evaluate the performance of the handover trigger in a real setting. Figure 11 shows the reports are combined and sent to the PBX. Figure 12 shows the target system into which we have integrated our handover trigger module. The server and terminal are from Optimobile[2] and comprises a system capable of voice roaming. The PBX connects to the local Ethernet *and* to the PSTN providing connectivity to the GSM network. We used an HP-6340 PDA terminal with 802.11 and GSM interfaces. Multi-path probing, by sending data over both interfaces simultaneously is not performed in this setup.

When evaluating the trigger performance, we need to match our objective score with the listening judgment of a test subject. The role of the test subject is to indicate at what time the 802.11 quality becomes unacceptable. Therefore, we called from the mobile terminal over the 802.11b network using VoIP via PBX to the public PSTN to a phone picking up constant speech. Using the 802.11b network, the test subject walked out of the office waiting for a handover to occur, the walk is shown in Figure 6. The handover was never performed, rather when the score fell below the chosen threshold the time was recorded in a file. Later we compared the trigger time with the time when the test subject indicated unacceptable quality. Ideally, the trigger time should precede the subjective time by five seconds since it requires approximately this time to establish a PSTN

[2] http://www.optimobile.se

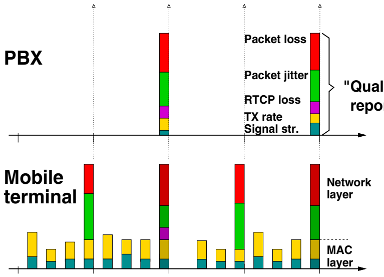


Fig. 11. Quality reports are sent periodically from the mobile to the PBX

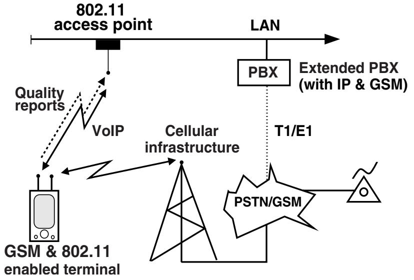


Fig. 12. The complete system used. Our module resides in the terminal & PBX.

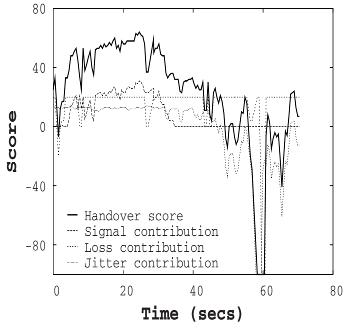


Fig. 13. Handover score when walking out of the office. The bold line is the score, the other lines its contributors.

Perceived quality started good *and became bad*	Timely HO **68**	Late HO 10
Perceived quality started good *and remained good*	Unnecessary HO 7	No HO **15**

Fig. 14. 100 trial handover (HO) results showing 83% success. The bold values show optimal decisions.

connection. Note that it is possible to subjectively judge whether the handover occurred too late i.e. perceived poor quality, however not too early unless one examines the recorded times.

Figure 13 shows the result of one coverage experiment whilst Figure 14 shows the results of 100 experiments. In most cases the quality did not deteriorate at the same physical location, due to radio interference and imperfect terminal software. In 68 cases the trigger released on time as desired. In 10 cases the trigger came too late, i.e. the subject perceived poor quality for a brief period while waiting for handover to occur. In 7 cases the trigger suggested an unnecessary handover, i.e. the call became more expensive than necessary. The remaining 15 runs never triggered handover which is optimal. Therefore in 83% of the cases the algorithm made the ideal decision. In 10% of the cases quality temporarily deteriorated because the handover came late, this is inconvenient but not fatal.

6 Related Work

Calvagna et al. present an overview of handover issues with a focus on hybrid mobile data networks [10]. They propose a neural network solution for handovers to/from 802.11 networks to GPRS networks and show its performance to be good. The E-Model as standardized by the ITU-T allows for the prediction of voice quality based on network QoS parameters [4]. However, it is not useful for our purposes because it does not take the signal strength and delay jitter into account. Very recent work by Hoene et al. propose a real-time implementation of PESQ called PESQlite [3]. It reduces the complexity by making simplifications to the PESQ algorithm e.g. using constant length test samples and non time alignment of the degraded samples. Our off-line method has a slightly different purpose, it is to obtain a mapping between consecutive packet loss and the PESQ MOS score. Dimitriou et al. state that interference and users moving out of range as limiting factors for good VoIP quality in WLANs [1]. Their solution is to use better speech coding and suggest an enhanced version of G.711 to make the speech more resilient to loss. Kashihara and Oie developed a WLAN handover scheme for VoIP that makes use of MAC-layer information on the number of retransmissions of the voice packets [11]. If this number exceeds a certain threshold, the system switches to multi-path transmission of the packets. As soon as one of the WLAN interfaces reaches a stable condition, it can be used for single-path transmission. In Fitzpatrick et al. propose a transport layer handover mechanism using the stream control transmission protocol (SCTP) [7]. The mechanism uses the multi-homing feature of SCTP and measures the network performance metrics by sending probes. Handover decisions are based on speech quality estimations utilizing the ITU-T's E-Model.

7 Conclusions, Future Work and Acknowledgments

The goal of this work was to map measurable parameters to speech quality in order to implement triggers for voice handovers. The solution was integrated into an existing system for evaluation. We have shown that automatic network roaming worked ideally in 83% of the trials we conducted. The results of the experiments can be changed by choosing the threshold value of the trigger. More precisely the balance between remaining in the 802.11 network longer and switching earlier can be chosen. Therefore the threshold value can be seen as a monetary selection. The fraction of expensive calls may be reduced by lowering the threshold but this will increase the periods of deteriorated quality. In the case where the mobile roams from the cellular to the 802.11 network, i.e. enters a LAN. A different approach is needed where probing the quality before handing over would be more appropriate. This work has been partly supported by the European Union under the E-Next Project FP6-506869, the Vinnova SIBED program in Sweden and the Austrian government's Kplus competence center program. We are very grateful to Optimobile AB for allowing us to use their system in the testing and evaluation phases. Thanks to Bengt Ahlgren, Pekka Hedqvist, Henrik Lundqvist, Per Gunningberg, Gunnar Karlsson, Martín Varela and Thiemo Voigt for their valuable comments on this paper.

References

1. E. Dimitriou and P. Sörqvist. Internet Telephony over WLANS. Technical report, Global IP Sound, Sept. 2003. http://www.globalipsound.com/solutions/wlan.usta_paper.pdf.
2. F. Hammer, P. Reichl, and T. Ziegler. Where Packet Traces meet Speech Samples: an Instrumental Approach to Perceptual QoS Evaluation of VoIP. In *IEEE International Workshop on Quality of Service IWQOS 2004*, pages 273–280, Montreal, Canada, June 2004.
3. C. Hoene. *Internet Telephony over Wireless Links*. PhD thesis, Technical University of Berlin, Germany, Dec. 2005.
4. International Telecommunication Union. The E-model, a computational model for use in transmission planning. Recommendation G.107, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Dec. 1998.
5. International Telecommunication Union. Appendix I: A high quality low-complexity algorithm for packet loss concealment with G.711. *ITU-T Recommendation G.711, Appendix I*, Sept. 1999.
6. International Telecommunication Union. Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. Technical report, Telecommunication Standardization Sector of ITU, Feb. 2001.
7. John Fitzpatrick and Sen Murphy and John Murphy. An Approach to Transport Layer Handover of VoIP over WLAN. In *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, USA, Jan. 2006.
8. A. Kamerman and L. Monteban. WaveLAN-II: A High-performance wireless LAN for the unlicensed band. *Bell Lab Technical Journal*, pages 123–140, Apr 1990.
9. M. Lacage, M. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *ACM International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, Venice, Italy, Oct. 2004.
10. K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Y. J. Mäkelä, R. P. R., and J. V. J. Handoff in hybrid mobile data networks. *IEEE Personal Communications Magazine*, pages 34–47, Apr. 2000.
11. Shigeru Kashihara and Yuji Oie. Handover Management based upon the number of retries for VoIP in WLANs. In *Proc. IEEE Vehicular Technology Conference (VTC2005)*, May 2005.

An Efficient Algorithm for Resource Sharing in Peer-to-Peer Networks

Wei-Cherng Liao, Fragkiskos Papadopoulos, and Konstantinos Psounis

University of Southern California, Los Angeles, CA 90089
{weicherl, fpapadop, kpsounis}@usc.edu

Abstract. The performance of peer-to-peer systems depends on the level of cooperation of the system's participants. While most existing peer-to-peer architectures have assumed that users are generally cooperative, there is great evidence from widely deployed systems suggesting the opposite. To date, many schemes have been proposed to alleviate this problem. However, the majority of these schemes are either too complex to use in practice, or do not provide strong enough incentives for cooperation.

In this work we propose a scheme based on the general idea that offering uploads brings revenue to a node, and performing downloads has a cost. We also introduce a theoretical model that predicts the performance of the system and computes the values of the scheme's parameters that achieve a desired performance. Our scheme is quite simple and very easy to implement. At the same time, it provides very strong incentives for cooperation and improves the performance of P2P networks significantly. In particular, theory and realistic simulations show that it reduces the query response times and file download delays by one order of magnitude, and doubles the system's throughput.

Keywords: P2P networks, user cooperation, theoretical analysis, realistic simulations.

1 Introduction

Peer-to-peer (P2P) systems provide a powerful infrastructure for large-scale distributed applications, such as file sharing. While cooperation among the system's participants is a key element to achieve good performance, there has been growing evidence from widely deployed systems that peers are usually not cooperative. For example, a well known study of the Gnutella file sharing system in 2000 reveals that almost 70% of all peers only consume resources (download files), without providing any files to the system at all [1]. This phenomenon is called "free-riding".

Despite the fact that this phenomenon was identified several years ago, recent studies of P2P systems show that the percentage of free-riders has significantly increased [2]. This is not because industry and academia have ignored the problem. There is a large body of work on incentive mechanisms for P2P networks, varying from centralized and decentralized credit-based mechanisms, *e.g.*

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 592–605, 2006.

[3, 4, 5, 6], to game-theoretic approaches and utility-based schemes, *e.g.* [7, 8], to schemes that attempt to identify and/or penalize free-riders, *e.g.* [9, 10, 11], the last two being proposed by the popular KaZaA and eMule systems. The problem of free-riders is hard to tackle because the solution has to satisfy conflicting requirements: minimal overhead, ease of use, and at the same time good amount of fairness and resilience to hacking.

In this paper we propose and study the performance of an efficient algorithm that is very easy to use, it enforces users to be fair, and it can be implemented in a number of ways that tradeoff overhead and resilience to malicious users. According to the algorithm, users use tokens as a means to trade bytes within the system. A user earns K_{up} tokens for each byte he/she uploads to the system and spends K_{down} tokens for each byte he/she downloads from the system. The user also gains K_{on} tokens for each second his/her machine is on the system (*i.e.* it is online). A user can initiate a download only if the number of tokens that he/she has is large enough to download the complete file.

The proposed algorithm relies on the general idea that users should be awarded for offering uploads and staying online, and pay for performing downloads. While others have proposed solutions that use the same general idea in the past, *e.g.* [6, 8], there are a number of questions that either have not been addressed at all or have been studied via simulations only: (i) How should one tune the parameters that dictate the gain from uploads and the cost of downloads? Specifically to our scheme, what is the right value for the parameters K_{on}, K_{up}, K_{down}? (ii) What is the exact effect of such an algorithm on overall system performance over a wide range of conditions? (iii) Would a *small* number of malicious users, that manage to subvert the scheme, degrade overall performance noticeably? (iv) Is it possible to trade off one performance metric for another by varying the parameters of the algorithm, e.g. trade off download delay for total system capacity? Our theoretical analysis of the performance of the resulting system, coupled with extensive realistic simulations, gives concrete answers to all these questions. Interestingly enough, it shows that the query response times and file download delays can be reduced by one order of magnitude while being able to sustain higher user download demands.

An important aspect of any solution to the free-riding problem is if the information about the users' behavior is determined and maintained locally and without any interaction with the other peers of the system (localized solutions), or it is determined and maintained by either a centralized authority (non-localized centralized solutions), or by the continuous exchange of information among the system's participants (non-localized distributed solutions). Localized solutions are simple and impose very little overhead but they are easy to subvert. Non-localized solutions are hard to subvert but are complex to use in practice. (KaZaA, eMule, and BitTorrent all use localized approaches.) In any case, our proposed scheme can be easily implemented in any way, and we describe later in the paper how to implement it efficiently with each one of the approaches.

The rest of the paper is organized as follows: In Section 2 we briefly discuss prior work on providing incentives for P2P systems. In Section 3 we provide a detailed description of the proposed scheme. In Section 4 we provide a mathematical model that predicts the performance of the system and gives guidelines on how to set the scheme's parameters. In Section 5 we present realistic experiments of P2P systems on top of TCP networks. In Section 6 we briefly discuss some implementation thoughts and finally conclude in Section 7.

2 Related Work

There has been a large body of work on incentive mechanisms for P2P networks. Three of the most popular localized schemes are the ones implemented by the eMule [11], the KaZaA [12], and the BitTorrent [13] systems. eMule rewards users that provide files to the system by reducing their waiting time when they upload files using a scoring function (called *QueueRank*). Similarly, in KaZaA, each peer computes locally its *Participation Level* as a function of download and upload volumes, and peers with high participation levels have higher priority [10]. A disadvantage of both of these schemes is that they provide relatively weak incentives for cooperation since peers that have not contributed to the system at all can still benefit from it, if they are patient enough to wait in the upload queues. Other problems include that they favor users with high access bandwidth, which may result in frustration or a feeling of unfairness [14], and that they are vulnerable to the creation and distribution of hacked daemons that do not conform to the desired behavior [15]. BitTorrent uses a different scheme that is specific to its architecture. Each peer periodically stops offering uploads to its neighbors that haven't been offering uploads to him/her recently. This scheme is hard to subvert. However, it suffers from some unfairness issues and it only works with "BitTorrent-style" systems, that is, in systems where files are broken into pieces, and downloading a file involves being connected to almost all of ones neighbors in order to collect and reassemble all the pieces of the file.

Non-localized proposals are primarily concerned with creating systems that cannot be subverted. Some of them make use of credit/cash-based systems. They achieve protection from hackers by either using central trusted servers to issue payments (centralized approach), *e.g.* [3, 4], or by distributing the responsibility of transactions to a large number of peers (distributed approach), *e.g.* [6]. Other distributed approaches use lighter-weight exchanged-based mechanisms, *e.g.* [16], or reputation management schemes, *e.g.* [5]. These mechanisms are indeed hard to subvert but they are also quite complex to use in practice [16].

In this paper we decouple the issue of how to design an algorithm to prevent free-riding from the issue of how to implement this algorithm in a P2P system. We first propose an efficient scheme that provides very strong incentives for cooperation. We show this via both theory and simulations. Then, we show that the scheme is generally applicable to any P2P system and comment on how to implement it using either a localized, or a non-localized approach. Another important contribution is the theoretical analysis of the performance of a P2P system with and without the proposed scheme. The analysis yields a set of equations that are used to predict the system's performance under a wide range of conditions, and to tune the parameters of the scheme.

3 A Simple and Effective Algorithm

As mentioned earlier, the algorithm uses tokens as a means to trade bytes within the system. Each user is given an initial number of tokens M when he/she first joins the network. This allows new users to start downloading a small number of files as soon as they join the system. When a user rejoins the system he/she uses the amount of tokens he/she previously had.

Users spend K_{down} tokens for each byte they download from the system and earn K_{up} tokens for each byte they upload to the system. This forces users to offer files for upload proportionally to the number of files they want to download. Further, users gain K_{on} tokens/sec while being online. This mechanism of accumulating tokens serves two purposes. First, it allows users who are not contacted frequently for uploads to gain tokens by just being online, which is more fair towards users with low access bandwidth [14]. Second, it provides an incentive for users to keep their machines on the system even when they are not downloading a file, which helps to prevent the so-called problem of "low availability" [17]. Note that the value of K_{on} should be relatively small, in order to prevent users from gaining many tokens by just keeping their machines on without providing any uploads. Finally, a user can initiate a download only if the number of tokens he/she currently possesses is greater or equal to the number of tokens required to download the requested file.

This scheme provides strong incentives for cooperation. Free-riders are "forced" to provide some uploads to the system in order to gain tokens fast enough to sustain their desirable download demands. Some free-riders may decide to share their files as soon as they are out of tokens. Others may adopt a more dynamic behavior and decide to adjust the number of uploads they provide to the system as a function of the number of tokens they currently have. In any case, the change in the free-rider's behavior increases the amount of available system resources tremendously, which, in turn, significantly improves the system's performance, as we shall see in Section 5.

4 A Mathematical Model for the Proposed Scheme

In this section we derive a mathematical model that predicts the system's performance and can be used to tune the parameters of the scheme. Predicting the performance of the system from the model is beneficial because the alternative is P2P simulations/experiments, and those either involve a significantly smaller number of peers than in reality, or are prohibitively expensive. Tuning the parameters of the scheme is important because an arbitrary setting of their parameters may lead to several undesired situations. For example, giving a large value to K_{on} may provide tokens to the free-riders fast enough, so that there won't be any reason for them to start sharing their files with the system. As another example, giving relatively small values to both K_{on} and K_{up} may reduce the token accumulation rate of cooperative users so much such that they can't sustain their download demands.

4.1 System Dynamics

We assume a system that implements the proposed scheme which we call "system with the tokens". Recall that K_{down} and K_{up} are expressed in tokens/byte and K_{on} in tokens/sec. Now, let C_{down} and C_{up} denote the file download and upload speeds of a user (access line bandwidth), both expressed in bytes/sec. The user spends $K_{down}C_{down}dt$ tokens if he/she is downloading files from other peers during time $(t, t+dt)$. Also, he/she earns $K_{on}dt$ tokens if he/she is online during time $(t, t+dt)$ and $K_{up}C_{up}dt$ tokens if other users are uploading files from the user under study during time $(t, t+dt)$. Let $T(t)$ denote the number of the user's tokens at time t, with $T(0) \geq 0$. We can then write the following differential equation:

$$\frac{dT(t)}{dt} = K_{on}I_{on}(t) + K_{up}C_{up}I_{up}(t) - K_{down}C_{down}I_{down}(t), \tag{1}$$

where

$$I_{on}(t) = \begin{cases} 1 & \text{if the user is online in (t, t+dt)} \\ 0 & \text{otherwise} \end{cases},$$

$$I_{up}(t) = \begin{cases} 1 & \text{if the user provides uploads in (t, t+dt)} \\ 0 & \text{otherwise} \end{cases},$$

$$I_{down}(t) = \begin{cases} 1 & \text{if the user performs downloads in (t, t+dt)} \\ 0 & \text{otherwise} \end{cases}.$$

Taking expectations on both sides of Equation (1), and interchanging the expectation with the derivative on the left hand side[1], we get:

$$\frac{dE[T(t)]}{dt} = K_{on}P_{on}(t) + K_{up}C_{up}P_{up}(t) - K_{down}C_{down}P_{down}(t), \tag{2}$$

where $P_{on}(t)$ is the probability the user is online at time t, $P_{up}(t)$ is the probability that the user provides uploads to the system at time t, and $P_{down}(t)$ is the probability that the user performs downloads from the system at time t. Note that Equation (2) can be regarded as a fluid model describing the token dynamics.

$P_{on}(t)$, $P_{up}(t)$, and $P_{down}(t)$ depend on how the user behaves given the number of tokens that he/she has at some point in time, and on his/her download

[1] Taking into account that $T(t)$ is bounded in practice, we can use the bounded convergence theorem [18] to justify the interchange.

demands. Along these lines, one can define user profiles and solve the differential equation. Due to limitations of space we will not proceed with this task here. (The interested reader is referred to [19]). Instead, we will only study the steady state by setting $\frac{dE[T(t)]}{dt} = 0$, and dropping the time dependence from the probabilities in Equation (2). Note that the existence of a steady state can be easily justified for a free-rider, by taking into consideration that in the long-run he/she will spend as many tokens as he/she gains. [2]

Without loss of generality assume $P_{on} = 1$. [3] Let R_{up} be the long-run average rate of file upload requests per second that the user handles, which we refer to as the *upload rate*. Also, let R_{down} be the long-run average rate of file download requests per second that the user initiates, which we refer to as the *download rate*. Last, let S denote the average file size in the system in bytes. Then, it is easy to see that $P_{up} = \frac{R_{up}S}{C_{up}}$ and $P_{down} = \frac{R_{down}S}{C_{down}}$. [4] Equation (2) in steady state yields:

$$K_{on} + K_{up}R_{up}S - K_{down}R_{down}S = 0.$$

Taking the average over all free-riders yields:

$$K_{up} = K_{down}\left(\frac{E[R_{down}|\text{FR}]}{E[R_{up}|\text{FR}]}\right) - \frac{K_{on}}{E[R_{up}|\text{FR}]S}. \qquad (3)$$

Equation (3) relates the parameters of the scheme, K_{on}, K_{up}, and K_{down}, with the average download and upload activity of free-riders. We will later use it to select the parameter values that yield a target performance. But first, we need to compute the average download and upload rates, which is the next topic.

4.2 User Download Rate (R_{down})

Let N be the number of peers in the system and let a proportion α of them be free-riders. Assume that free-riders are uniformly distributed over the system. Also, assume that both cooperative users and free-riders have the same download demands. In particular, they have the same query request rate, denoted by R_q queries/sec, and the same preference over files, that is, each query is for file i with some probability $Q_f(i)$ irrespectively of the query issuer. Finally, assume that free-riders respond to a query only if the amount of tokens they currently have is less than the amount required to download the file they currently desire. (Recall that cooperative users always respond to query requests.)

Let $P_{ans}(i)$ be the probability that a query request for file i is successfully answered. Now, recall that in the system with the tokens a user can initiate a download only if the amount of tokens he/she has is larger than the amount

[2] Considering the existence of a steady state for a non-freerider is a bit more involved. As we will shortly see he/she may or may not have a steady state. Nevertheless, this will not be important for the system dynamics.

[3] A similar analysis holds for $P_{on} < 1$.

[4] Assuming a stable system, the exact values of C_{up} and C_{down} are not important for our analysis.

required to download the file. Let P_{tkn}^{FR} and P_{tkn}^{NF} denote respectively the probability that a free-rider and a non-freerider have enough tokens to initiate a download. Then, we can express the average download rate of free-riders and non-freeriders as follows:

$$E[R_{down}|\text{FR}] = \sum_i R_q \cdot Q_f(i) \cdot P_{ans}(i) \cdot P_{tkn}^{FR}, \tag{4}$$

$$E[R_{down}|\text{NF}] = \sum_i R_q \cdot Q_f(i) \cdot P_{ans}(i) \cdot P_{tkn}^{NF}, \tag{5}$$

where the summation is taken over all files i. Clearly, the average download rate over all users in the system is:

$$E[R_{down}] = E[R_{down}|\text{FR}] \cdot \alpha + E[R_{down}|\text{NF}] \cdot (1 - \alpha). \tag{6}$$

To complete the calculation of the download rates, note that R_q, $Q_f(i)$, and α are given quantities. (There exist a large body of work in measurement studies of P2P systems, e.g. [20, 21], from which one can deduce typical values for these quantities.) Hence, what remains is to compute P_{ans}, P_{tkn}^{FR}, and P_{tkn}^{NF}. We start by deriving a relation between P_{tkn}^{FR} and P_{tkn}^{NF}. First, note that in steady state the token earning rate equals the token spending rate for each free-rider. A free-rider responds to a query request only when he/she doesn't have enough tokens, i.e. with probability $1 - P_{tkn}^{FR}$. Since a non-freerider always responds to a query request, it is easy to see that the token earning rate of free-riders over that of non-freeriders equals $1 - P_{tkn}^{FR}$. Now, the token spending rate is proportional to the download rate, and Equations (4) and (5) imply that the token spending rate of free-riders over that of non-freeriders equals $\frac{P_{tkn}^{FR}}{P_{tkn}^{NF}}$. Assuming that non-freeriders are also in steady state (in which case Equation (3) also holds if the average is taken with respect to non-freeriders only), we can equate the two ratios and write $P_{tkn}^{NF} = \frac{P_{tkn}^{FR}}{1-P_{tkn}^{FR}}$. Clearly, this equality is valid for $P_{tkn}^{FR} \leq 0.5$. In particular when $P_{tkn}^{FR} = 0.5$, $P_{tkn}^{NF} = 1$, which implies that non-freeriders always have enough tokens to initiate downloads. For $P_{tkn}^{FR} > 0.5$, the last equality no longer holds. In this case the token earning rate of non-freeriders will be larger than their token spending rate, which implies that their amount of tokens will continuously increase. However, this still suggests that $P_{tkn}^{NF} = 1$. We can now write:

$$P_{tkn}^{NF} = \min\left(1, \frac{P_{tkn}^{FR}}{1 - P_{tkn}^{FR}}\right). \tag{7}$$

Now, lets find a relation for $P_{ans}(i)$. First, assume that due to congestion at the overlay layer [22], each message (either a query request or a query response) has a probability p of being dropped at some peer. [5] Then, if L is the average

[5] This assumption is introduced to make the model more general. A well designed system usually has $p \approx 0$, which is accomplished by setting the buffer size of the TCP socket sufficiently large.

number of overlay hops until a query is answered, $P_{drop} = 1 - (1-p)^L$ is the probability that the query response is lost. Next, observe that if $K \leq N$ is the average number of peers that a query request can reach, then the request can be answered by an average of $K \cdot ((1 - P_{tkn}^{FR}) \cdot \alpha + 1 \cdot (1-\alpha))$ peers. Finally, let $P_f(i)$ be the probability that a peer has file i. We can then write:

$$P_{ans}(i) = 1 - (1 - P_f(i) \cdot (1 - P_{drop}))^{K \cdot ((1 - P_{tkn}^{FR}) \cdot \alpha + 1 - \alpha)}. \tag{8}$$

4.3 User Upload Rate (R_{up})

The total number of downloads equals the total number of uploads, and thus the expected download and upload rates over *all* nodes are also equal. This does not mean that all peers provide uploads. For example, in a system that does not implement the proposed scheme $E[R_{down}] = E[R_{up}]$ but we know that only non-freeriders provide uploads, i.e. $E[R_{up}|\text{FR}] = 0$, and hence $E[R_{up}|\text{NF}] = \frac{E[R_{down}]}{(1-\alpha)}$. On the other hand, in the system with the tokens each free-rider answers to a query request with probability $1 - P_{tkn}^{FR}$. As a result, this system behaves as if there are $N \cdot ((1-\alpha) + \alpha \cdot (1 - P_{tkn}^{FR}))$ non-freeriders. It is easy to see that the expected upload rate of each non-freerider is now given by:

$$E[R_{up}|\text{NF}] = \frac{E[R_{down}]}{(1-\alpha) + \alpha \cdot (1 - P_{tkn}^{FR})}. \tag{9}$$

And, since $E[R_{up}] = E[R_{down}]$, the expected upload rate of each free-rider equals:

$$E[R_{up}|\text{FR}] = \frac{(1 - P_{tkn}^{FR}) \cdot E[R_{down}]}{(1-\alpha) + \alpha \cdot (1 - P_{tkn}^{FR})}. \tag{10}$$

4.4 Choosing the Right Values for K_{on}, K_{up}, and K_{down}

We use P_{tkn}^{FR} as the design parameter of our system since it dictates how often free-riders offer uploads, which, in turn, specifies the average amount of available resources in the system. We are given the query- and file-popularity probability functions $Q_f(i)$, $P_f(i)$, the query request rate R_q, and information about the overlay network. (For example, information about the overlay network includes the percentile of free-riders α, the socket buffer sizes that dictate the drop probability p, and the structure of the overlay graph as well as the search algorithm that dictate the number of peers that a query reaches K and the average path length between a query issuer and a query responder L.) We want to find a set of values for K_{on}, K_{up} and K_{down} that will satisfy a target P_{tkn}^{FR}, and, in turn, a target performance.

First, observe from Equation (3) that it is the *relative* values of K_{on}, K_{up}, and K_{down} that are important for the proper operation of the system. Recall also that K_{on} should be sufficiently smaller than the token spending rate of free-riders. This is to prevent them from accumulating enough tokens by just staying online without offering any uploads. Thus, we should have $K_{on} \ll K_{down}E[R_{down}|\mathrm{FR}]S$.

With the above observations in mind we proceed as follows in order to satisfy the target P_{tkn}^{FR}:

(i) Fix K_{down} to some arbitrary value,
(ii) use Equation (7) to compute P_{tkn}^{NF}, (To guarantee that cooperative users will not be penalized, P_{tkn}^{NF} should be close to 1.)
(iii) use Equations (4) and (8) to compute the value of $E[R_{down}|\mathrm{FR}]$, and Equations (10), (6) and (5) to compute $E[R_{up}|\mathrm{FR}]$,
(iv) assign a value to K_{on} which is one order of magnitude smaller than $K_{down}E[R_{down}|\mathrm{FR}]S$, (The specific value turns out not to affect the performance sizeably.) and
(v) use Equation (3) to find the right value for K_{up}.

Conversely, if we are given the values of K_{on}, K_{up}, and K_{down} we can use our equations to predict quantities like $E[R_{down}|\mathrm{FR}]$, $E[R_{down}|\mathrm{NF}]$, $E[R_{up}|\mathrm{FR}]$ and so on. [6]

In the next Section we verify the accuracy of our analysis via experiments on top of TCP networks, and show the impact of the proposed scheme on system's performance.

5 Experiments

5.1 Simulation Setup

For our experiments we use GnutellaSim [23], a packet-level peer-to-peer simulator build on top of ns-2 [24], which runs as a Gnutella system. We implement the file downloading operation using the HTTP utilities of ns-2.

We use a 100-node transit-stub network topology as the backbone, generated with GT-ITM [25]. We attach a leaf node to each stub node. Each leaf node represents a peer. The propagation delays assigned to the links of the topology are proportional to their length and are in the order of ms. We assign capacities to the network such that the congestion levels are moderate. The capacity assigned to a peer's access link is 1.5Mbps.

In order to test the algorithm on a general gnutella-like unstructured P2P network we use Gnutella v0.4, which uses pure flooding as the search algorithm and does not distinguish between peers. The TTL for a query request message is set to 7 (the default value used in Gnutella).

All peers join the system initially and never go offline. For simulation purposes we implement the following user behavior: each user initiates query requests at the constant rate of 1 query every 20sec. Once a timeout for a query request occurs, the corresponding query is retransmitted. The maximum number of retransmissions is set to 5, and the timeout to 60sec.

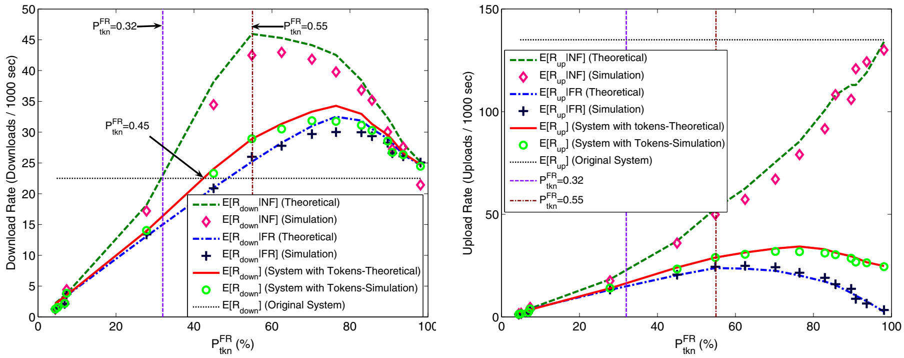
[6] Note that we can also use Equations (4)...(10) to compute upload/download rates in a system that does not implement the scheme, by setting $P_{tkn}^{FR} = 1$.

There are 1000 distinct files in the system, $i = 1...1000$. A query request is for file i with probability proportional to $\frac{1}{i}$ (Zipf distribution). The number of replicas of a certain file is also described by a Zipf distribution with a scaling parameter equal to 1, and the replicas of a certain file are uniformly distributed among all peers. These settings are in accordance with measurement studies from real P2P networks [20, 21]. We distinguish two systems: (i) the original system which does not implement the proposed algorithm, and (ii) the system with the tokens. In both systems, 85% of peers are free-riders in accordance to the percentage reported in [2]. Finally, the file size is set to 1MB.

5.2 Simulation Results

Download and Upload Rates. For various values of the design parameter P_{tkn}^{FR} we compute the corresponding values of K_{on}, K_{up} and K_{down} according to the procedure described in the previous Section. We then assign these values to all users of the system and compare the theoretical download and upload rates with the experimental results. Figures 1(i) and 1(ii) show respectively the expected download and upload rate over all non-freeriders, over all free-riders, and over all users of the system, as a function of P_{tkn}^{FR}. The horizontal line in Figure 1(i) represents the expected download rate of a user in the original system. (Clearly, in the original system $E[R_{down}] = E[R_{down}|FR] = E[R_{down}|NF]$.) The horizontal line in Figure 1(ii) represents the expected upload rate of a non-freerider in the original system. (Recall that in this system $E[R_{up}|FR] = 0$.)

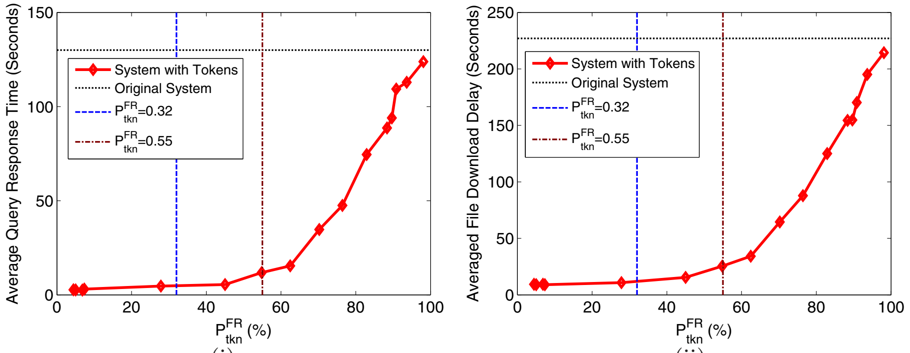
It is clear from the plots that analytical and simulation results match. Further, we can make several interesting observations. First, notice that as P_{tkn}^{FR} increases, the download rate for both classes of users first increases and then starts decreasing until it reaches the value of the original system. Second, observe that while the upload rate of free-riders behaves in a similar manner, the upload rate of non-freeriders continuously increases until it reaches its original value. Based on these observations we divide the plots into three regions. The



(i) (ii)

Fig. 1. (i) User's expected download rate, and (ii) user's expected upload rate first region corresponds to $P_{tkn}^{FR} < 0.32$. In this region, both classes of peers are constrained to a lower download rate compared to the original system, since the probability of having tokens to initiate a new download after a successful query is pretty low. Notice that for $P_{tkn}^{FR} = 0.32$, and hence for $P_{tkn}^{NF} = 0.47 < 1$, cooperative users can at least sustain the same download rate they had in the original system. The second region corresponds to $0.32 \leq P_{tkn}^{FR} \leq 0.55$. In this region, users accumulate tokens at a higher rate than before. Since there are more responses than in the original network, users can use the extra tokens to initiate more downloads. Notice that cooperative users earn tokens faster than free-riders since they always respond to query requests. At $P_{tkn}^{FR} = 0.55$, non-freeriders achieve their maximum download rate, which is approximately twice the one they had in the original system. Finally, the third region corresponds to $0.55 < P_{tkn}^{FR} \leq 1$. In this region free-riders accumulate tokens faster than before and reduce their query response rate since they do not need to provide as many uploads as before. This causes cooperative users to handle more uploads. Futher, since the query response rate regulates the download rate, the latter also decreases. At $P_{tkn}^{FR} = 1$, the two systems have approximately the same performance, as expected.

Impact on Delays. Figures 2(i) and 2(ii) show respectively the average query response time (that includes retransmissions) and the average download delay as a function of P_{tkn}^{FR}. The plots can be divided in the same three regions as before. For $P_{tkn}^{FR} < 0.32$, the low user download rate imposes a low load into the network. This yields the low delays. For $0.32 \leq P_{tkn}^{FR} \leq 0.55$, as the user download rate increases, the load in the network and hence the delays also increase. Note that the query and download delays are still significantly smaller than in the original system, despite that the download rate, and hence the load, is higher. This is because a significant portion of the load is now handled by the free-riders. For $0.55 < P_{tkn}^{FR} \leq 1$ the delays continue to increase even though the download rate decreases. This is because free-riders provide fewer and fewer uploads. As P_{tkn}^{FR}



(i) (ii)

Fig. 2. (i) Average query response time, and (ii) average file download delay approaches 1, the performance of the two systems is approximately the same. To fairly compare the delays between the two systems, we should consider the case where the load is the same, i.e. where $E[R_{down}] = 22$ downloads/1000sec. This value corresponds to $P_{tkn}^{FR} = 0.45$, and as we can see from the plots this corresponds to approximately one order of magnitude lower query and file download delays. This is a gigantic amount of improvement on the system's performance.

As a final note, the best operating region is the second, where $0.32 \leq P_{tkn}^{FR} \leq 0.55$. In this region, we can either choose to operate the system at $P_{tkn}^{FR} = 0.32$, where cooperative users can sustain the same download demands as in the original system, or sacrifice a bit from the performance improvement with respect to reduced delays to support higher user demands.

6 Implementing the Scheme

As mentioned before, this scheme can be implemented either locally or non-locally. Implementing this scheme locally is quite simple. The local P2P client takes care of bookkeeping by increasing the user's tokens for each acknowledged byte he/she uploads and for being online, and by decreasing the tokens for each byte the user downloads. However as we have already mentioned, this approach is quite vulnerable to hacked clients.

There are several directions for making the hacking of localized solutions hard. One can utilize encryption techniques *e.g.* [26] that make unauthorized modifications to data (such as the scheme's parameters) hard. In addition, one can also use technologies like DRM (Digital Rights Management) in order to protect the entire client's code from being altered *e.g.* [27], and re-distribute new clients frequently in order to minimize the number of hacked clients that can be connected to the network. Further, one could also employ techniques such as tamper-proofing and self-checking in order to verify the client's code integrity during the join process and/or on every download request *e.g.* [28, 29, 30, 31]. Of course, the only way to guarantee that all P2P clients are original is to have a trusted platform where both the hardware and the operating system can be trusted [32]. This is clearly not an option in practice. However, interestingly enough, both theory and simulations dictate that our scheme is quite resilient to a *small* number of hacked clients. In particular, the system performance is virtually unchanged when the hackers comprise less than 10% [19]. Hence, all one needs to do is to make it hard for users to use hacked clients.

The scheme can be also implemented in a secure non-localized centralized manner, where peers exchange messages with a centralized trusted authority that updates and maintains their amount of tokens. Peers would communicate with the centralized authority once they finish downloading to report the source node and the file size, periodically while being online, and to get permission to initiate a new download. This is similar to the main idea that many centralized "cash-based" systems, *e.g* [3, 4], follow. Finally, the scheme can be also implemented in a secure non-localized decentralized manner, *e.g.* by utilizing the framework suggested in [6].

7 Conclusion

In this paper we studied a simple algorithm that provides strong incentives for cooperation in file sharing P2P networks. We derived a mathematical model that describes the system's dynamics and which can be used for parameter tuning and performance prediction. We demonstrated the effectiveness of the algorithm via experiments with TCP networks. Future work consists of performing larger scale experiments, implementing the scheme in an operational P2P network, and extending our analytical methodology to compute other important performance metrics, e.g. the improvement on the expected download delays and response times.

References

1. E. Adar and B. Huberman, "Free riding on gnutella," http://www.firstmonday.dk/issues/issue5_10/adar, October 2000 (accessed Aug. 2005).
2. D. Hughes, G. Coulson, and J. Walkerdine, "Free riding on gnutella revisited: the Bell Tolls?," *IEEE Distributed Systems Online Journal*, vol. 6, no. 6, June 2005.
3. "Mojonnation," http://www.mojonation.net/Mojonation.html (accessed Aug. 2005).
4. J. Ioannidis, S. Ioannidis, A. Keromytis, and V. Prevelakis, "Fileteller. paying and getting paid for file storage," in *Proc. of 6th International Conference on Financial Cryptography*, March 2002.
5. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "EigenRep: Reputation management in P2P networks," in *Proc. of 12th International World Wide Web Conference (WWW 2003)*, May 2003.
6. V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer, "KARMA: A secure economic framework for P2P resource sharing," in *1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.
7. C. Buragohain, D. Agrawal, and S. Suri, "A game-theoretic framework for incentives in P2P systems," in *Proc. of International Conference on Peer-to-Peer Computing*, Sep 2003.
8. L. Ramaswanmy and L. Liu, "Free-riding: A new challenge to peer-to-peer file sharing systems," in *Proc. of the 36th Hawaii international conference on system sciences*, 2003.
9. M. Feldman, C. Papadimitriou, I. Stoica, and J. Chuang., "Free-riding and whitewashing in Peer-toPeer systems," in *SIGCOMM Workshop*, 2004.
10. "KaZaA participation level," http://www.kazaa.com/us/help/glossary/participation_ratio.htm (accessed Aug. 2005).
11. "The emule project," http://www.emule-project.net/ (accessed Aug. 2005).
12. "KaZaA media desktop," http://www.kazaa.com/ (accessed Aug. 2005).
13. "Bittorrent," http://www.bittorrent.com/protocol.html (accessed Aug. 2005).
14. H. Bretzke and J. Vassileva, "Motivating cooperation in peer to peer networks," in *Proc. of User Modeling UM03*, June 2003.
15. "Hack KaZaA participation level," http://www.davesplanet.net/kazaa/ (accessed Aug. 2005).
16. K. Anagnostakis and M. Greenwald, "Exchanged-based incentive mechanisms for peer-to-peer file sharing," in *Proc. of 24th International Conference on Distributed Computing Systems*, 2004.

17. R. Bhagwan, S. Savage, and G. M. Voelker, "Understanding availability," in *Proc. of 2nd IPTPS*, 2003.
18. R. Durrett, *Probability: Theory and Examples*, Duxbury Press, 2nd edition, 1996.
19. W.-C. Liao, F. Papadopoulos, and K. Psounis, "An efficient algorithm for resource sharing in peer-to-peer networks," Tech. Rep. CENG-2005-15, University of Southern California, 2005.
20. S. Saroiu, K. P. Gummadi, R. J. Dunn, S.D. Gribble, and H. M. Levy, "An analysis of internet content delivery systems," in *Proc. of the Fifth Symposium on Operating System Design and Implementation (OSDI)*, December 2002.
21. J. Chu, K. Labonte, and B. N. Levine, "Availability and locality measurements of peer-to-peer file sharing systems," in *Proc. of SPIEITCom: Scalability and Traffic Control in IP Networks*, July 2002.
22. Mostafa Amar Qi He, "Congestion control and massage loss in gnutella networks," in *Proc. of Multimedia Computing and Networking*, 2004.
23. "Packet-level Peer-to-Peer Simulation Framework and GnutellaSim," `http://www.cc.gatech.edu/computing/compass/gnutella/` (accessed Oct. 2005).
24. "Network simulator," `http://www.isi.edu/nsnam/ns` (accessed Sep. 2005).
25. K. Calvert, M. Doar, and E. W. Zegura, "Modeling internet topology," *IEEE Communications Magazine*, 1997.
26. "Data encryption standard," `http://www.itl.nist.gov/fipspubs/fip46-2.htm` (accessed Oct. 2005).
27. T. Sander, *Security and Privacy in Digital Rights Management*, Springer, 1st Edition, 2002.
28. D. Aucsmith, "Tamper resistant software: An implementation," in *Proc. 1st International Information Hiding Workshop*, May 1996.
29. H. Chang and M. Atallah, "Protecting software code by guards," in *Proc. of 1st ACM Workshop on Digital Rights Management*, May 2002.
30. Y. Chen, R. Venkatesan, M. Cary, R. Pang, S. Sinha, and M. Jakubowski, "Oblivious hashing: A stealthy software integrity verification primitive," in *Proc. of 5th International Information Hiding Workshop*, October 2002.
31. C.S. Collberg and C. Thomborson, "Watermarking, tamper-proofing, and obfuscation - tools for software protection," *IEEE Transactions on Software Engineering*, vol. 28, no. 6, June 2002.
32. S. W. Smith, *Trusted Computing Platforms: Design and Applications*, Springer, 1st Edition, 2004.

On the Identification and Analysis of P2P Traffic Aggregation[⋆]

Trang Dinh Dang, Marcell Perényi, András Gefferth, and Sándor Molnár

High Speed Networks Laboratory,
Dept. of Telecommunications & Media Informatics,
Budapest University of Technology & Economics,
H-1117, Magyar tudósok krt. 2, Budapest, Hungary
{trang, perenyim, geffertha, molnar}@tmit.bme.hu

Abstract. The main purpose of this paper is twofold. First, we propose a novel identification method to reveal P2P traffic from traffic aggregation. Our method is based on a set of heuristics derived from the robust properties of P2P traffic. We show the high accuracy of the proposed algorithm based on a validation study. Second, several results of a comprehensive traffic analysis, focusing on the differences between P2P and non-P2P traffic, are reported in the paper. Our results show that the unique properties of P2P application traffic seem to fade away during aggregation and characteristics of the traffic will be similar to that of other non-P2P traffic aggregation.

1 Introduction

From the beginning of the new millennium the Internet traffic characteristics show a dramatic change due to the emerging *Peer-to-Peer* (P2P) applications. Starting from the first popular one (Napster) a number of new P2P based multimedia file sharing systems have been developed (FastTrack, eDonkey, Gnutella, Direct Connect, etc.). The traffic generated by these P2P applications consumes the biggest portion of bandwidth in campus networks, overtaking the traffic share of the world wide web [24, 2]. A common feature in all of these P2P applications is that they are built on the P2P system design where instead of using the server and client concept of the web each peer can function both as a server and a client to the other nodes of the network. This principle involves the adapting nature of P2P systems as individual peers join or leave the network. Another common feature of these P2P systems is that they are mainly used for multimedia file sharing (movies, music files, etc.), which frequently contain very large files (megabytes, gigabytes) in contrast to the typical small size of web pages (kilobytes).

A number of studies have been published in the field of P2P networking. Papers [1, 2, 3, 4] focus on the measurement of different P2P systems like Napster, Gnutella, KaZaA, and the traffic characterization and analysis of P2P traffic

[⋆] The research was supported by Ericsson Traffic Lab and Magyar Telekom Ltd., Hungary.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 606–617, 2006.

providing some interesting results of resource characteristics, user behavior, and network performance. Several analytic efforts to model the operation and performance of P2P systems have been presented so far. Queueing models are applied in [5, 6], while in [7, 8, 9] branching processes and Markov models are used to describe P2P systems in the early transient and steady state. P2P analysis using game theory is presented in [19] among others. Other studies, e.g. [10, 11, 12, 13], are concerned with the effective performance and the QoS issues of P2P systems. In addition, many papers [14, 15, 16, 17, 18] indicate various possible applications using P2P principles. Further approaches propose structured P2P systems using Distributed Hash Table (DHT) with several implementations like Pastry, Tapestry, CAN, Chord [20].

The P2P traffic characteristics are not fully explored today and there is a tendency that they will be even more difficult to analyze. In contrast to the first generation P2P systems the recent popular P2P applications disguise their generated traffic resulting in the problematic issue of *traffic identification*. The accurate P2P traffic identification is indispensable in traffic blocking, controlling, measurement and analysis. However, the issue is touched upon in only a few papers and the proposed solutions still have some drawbacks. The problem is that P2P communications are continuously changing, from TCP layers using well-known ports in some first versions to both TCP/UDP with arbitrary and/or jumping ports nowadays. A robust and accurate P2P traffic identification is vital for network operators and researchers but today there is a lack of published results on this field and this is our main motivation for the work presented in this paper.

The workload characteristics of peers participating in some P2P systems has been examined in several papers as mentioned before. However, from the aspect of service providers only little useful information can be gained from these studies. The service providers are less interested in the detailed activities of some particular P2P softwares but the traffic generated by peer users. This paper, in contrast, concentrates on those factors and characteristics of P2P communications which have an impact on the P2P traffic aggregation.

The rest of the paper is organized as follows. We describe our measurements and the pre-processed data in Section 2. Section 3 presents our heuristic P2P identification method. The traffic characterization results are given in Section 4. Finally, Section 5 concludes the paper.

2 Traffic Measurements

The measurements were taken at one of the largest Internet providers in Hungary in May 2005. In the chosen network segment, traffic of ADSL subscribers is multiplexed in some DSLAMs before entering the ATM access network. Placed at the border of the access network and the core network are some Cisco routers. NetFlow measurements are carried out at two of these routers in three days from May 26th to 28th. NetFlow, developed by Cisco, collects all *incoming* flow information and exports the logs periodically. The obtained data traces are the aggregate incoming traffic of more than 1000 ADSL subscribers.

Table 1. Summary of the collected data sets

Data sets	05_26	05_27	05_28
Time of measurement	05/26 17h15 - 05/27 7h	05/27 17h06 - 24h	05/28 0h - 24h
Number of flows	4 293 394	5 858 756	17 224 625
Total traffic [GB]	113.91	126.06	316.91

Three data sets were selected for analysis, which are denoted by 05_26, 05_27, and 05_28. The summary of the data sets is presented in Table 1.

3 P2P Traffic Identification

A number of published papers have dealt with the issue of P2P traffic identification. Port-based analysis is presented in [25]. [22, 26] provide a method using the relationships between P2P flows or P2P client/server. Identification based on application signatures is shown in [21, 23]. In addition, [23] also proposes another method of identification based on some heuristics. In summary, P2P traffic identification has two promising approaches:

- P2P traffic identification based on payload information
- P2P traffic identification based on flow dynamics

The first method can provide very high detection accuracy in case of well-known open P2P protocols. It takes advantage in the investigation of some named P2P systems. Its drawbacks appear in high processor claim (for payload check), and the continuous change of P2P protocols, which are not available in most of the cases. Moreover, it also raises a number of legal and privacy problems. The second one is simpler to perform but it implies heuristic methods yielding less accurate results. However, it does not depend directly on actual P2P systems, thus it is more consistent and suitable for the analysis of P2P traffic aggregation. In this paper we have chosen the second approach and present an accurate and robust simple P2P traffic identification method.

3.1 A Heuristic Method for P2P Traffic Identification

Our proposed heuristic method consists of six steps, each being associated with a group of P2P flows to be identified. At the beginning we try to classify a set of widely used Internet applications (except P2Ps) based on well-known port analysis.

Initial step. While port based analysis is less accurate to identify P2P traffic, it is still appropriate to distinguish traffic of common applications. Our exhausted search of these applications and their communication ports, in both TCP and UDP layers, results in a table of application ports. Flows with these ports in the *source_port* or *dest_port* are first extracted from the data sets. HTTP/SHTTP ports are not among these. The reason is that HTTP ports are not only used for web surfing but also by some P2P applications, e.g. KaZaA. The separation of web and P2P traffic is considered by the second heuristics.

Step 1. The first heuristics is based on the fact that many P2P protocols, e.g. eDonkey, Gnutella, Fasttrack, etc., use both TCP and UDP transport layers for communication. Reasonably the unreliable UDP is often used for control messaging, queries, and responses while data transmission relies on TCP. However, the large volume of UDP traffic observed in our measurement data indicates that UDP could also be used for data transfer. Thus by identifying those IP pairs which participate in concurrent TCP and UDP connections we can state that the traffic between these IP pairs is almost surely P2P.

This heuristics is similar to what is proposed in [23] with a little difference. We note that some other common applications like NETBIOS, DNS also utilize both TCP and UDP. [23] needs a post-processing to extract this kind of traffic from the result of the heuristics. In contrast, this is not necessary in our case since we have already done this in the initial step: these applications are among the common ones.

Step 2. The second heuristics tries to separate web and P2P traffic from flows using HTTP/SHTTP ports, i.e. 80, 8080, 443, etc. The typical difference between P2P and web communication of two hosts can be observed. In general, web servers use multiple parallel connections to hosts in order to transfer web pages text and images (also music, video contents in some cases). In contrast, data transmission between peers consists of one or more consecutive connections, i.e. only a single connection can be active at a time. This property is used to identify web servers and then the traffic originating from them.

The traffic using HTTP ports is divided into groups of individual IP pairs. The web server is the one with the IP address in the HTTP ports side which has parallel connections to its pair. We also differentiate between two cases: if the IP address of the web server belongs to the outside IP domain it is likely to be a public web server. Then all the HTTP traffic from them is marked as web traffic. In the other case only parallel flows with HTTP ports are marked as web traffic. The rest of this traffic group is P2P traffic.

Step 3. In the next step, P2P traffic is selected using default ports of P2P applications. P2P software often defines default ports for communication. It is true that in most cases peer users can change it to any arbitrary port (but it is not frequent since peer-to-peering is usually not prohibited for home users) or port can be dynamically chosen automatically or when firewall or port-blocking is observed. This step cannot detect all P2P connections, but once the traffic is collected we can be almost sure that it is from those concerned P2P systems.

A table of well-known ports used by some popular P2P applications is collected for this step. Flows containing these values in *source_port* or *dest_port* are all marked P2P.

Step 4. In normal TCP/UDP operation, at least one of the two ports is selected arbitrarily. It is not likely that flows with similar flow identities (*source_IP, dest_IP, source_port, dest_port, prot_byte, tos*) exist in relatively short measurements. This happens, however, in the case of P2P connections, if both source

and destination peers dedicate a fixed port for data transfer. File download of a file is often executed in several smaller chunks. Therefore multiple flows with the same flow identities can be generated by P2P software. This is the basis of this heuristics: the identical flows are from P2P applications if at least two of each are found.
Step 5. For the same reason as the above heuristics, it is not probable that a host (IP) will repeatedly choose a given arbitrary port for TCP/UDP connections unless it is a server. Web servers and other common server traffic is extracted by the previous heuristics, thus it is safe to introduce the next heuristics: if an IP uses a TCP/UDP port more than 5 times in the measurement period that {IP,port} pair indicates P2P traffic. The selected upper threshold (5) is a rule of thumb established empirically.
Step 6. The last heuristics is based on the fact that objects of P2P downloads often have large sizes from several MB in case of music files or smaller applications to hundreds of MB in case of video files and larger software packages. In addition, peer users are patient. P2P downloadings can last some ten minutes or hours. By this heuristics those flows are considered P2P flows which have flow size larger than 1 MB or flow length is longer than 10 minutes.

4 Analysis Studies

In this section, we first verify the robustness of the proposed P2P traffic identification method. Next the results of the identification of the measured data sets are presented. The detailed comparison analysis of P2P and non-P2P traffic aggregation follows.

4.1 Verification of the Identification Method

In order to examine the robustness of the heuristics presented in Section 3.1 a validation measurement was carried out. In this measurement besides gathering general and aggregated information of the traffic flows we also recorded the name of the corresponding application. This enabled us to validate the correctness of the proposed P2P traffic identification method.

The measurement collected the traffic generated by two Linux PCs running SMTP and web servers among others (although with very light traffic), and some P2P applications: *qtorrent*, *valknut*, and *aMule*. These are the Linux versions of the Bittorrent, Direct Connect and eDonkey systems, respectively. To challenge the identification method, we used non-default P2P ports. Several downloads were initiated, while the P2P clients were also enable to serve requests of other peers. The measured trace contains more than 120000 data flows.

We present the performance of each heuristics and the overall identification process in Fig. 1. The hit rate of each heuristics, counted in percentage, is the ratio of the number of *correctly marked* flows and the total number of *marked* flows by the heuristics. We note that the hit rate of the 6th heuristics is not shown in the figure because it marked no flows in this data set. The last two

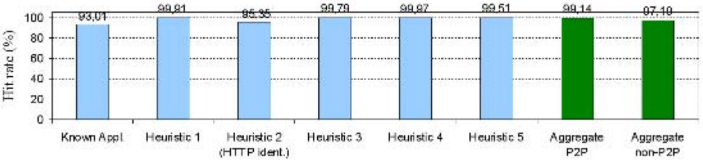


Fig. 1. Validation result of the identification method

columns in the figure show the rate of correctly marked P2P (non-P2P) flows and the total number of P2P (non-P2P) flows in the data set. The result is very convincing in every statistics. The average hit rate is greater than 99.7%. The amount of unidentified traffic is about 0.1%. The ratio of wrongly marked P2P flows and unidentified P2P flows per the total marked P2P flows are 0.3% and 0.8%, respectively. Note that these performance parameters are counted flow-wise. Similar results concerning the traffic quantities (bytes) are much better.

4.2 Traffic Identification

As described earlier the traces are the sets of flow information collected using Cisco NetFlow measurements (see Section 2). We assign a flag to each flow record of our database. The flag has the default value of u which means unknown (traffic) and it can be changed in the course of the identification process. The list of possible values of the flag is the following:

u: default value, unchanged if the flow cannot be classified
m: management flow (classified by IP addresses of the routers)
o: other non-TCP/UDP flow (ICMP, IPv6, RSVP, etc.)
k, kh: known common application (except HTTPs), flow using HTTP ports
pX: P2P flow, X denotes the heuristics which identifies the flow

The result of the identification procedure is summarized in Table 2.

Table 2. Traffic identification result

Data sets	05_26		05_27		05_28	
Flag	#flows [%]	volume [%]	#flows [%]	volume [%]	#flows [%]	volume [%]
m	0.5	0.01	0.08	0.005	0.1	0.007
o	0.42	0.06	0.87	0.05	0.88	0.29
k, kh	64.75	52.61	33.66	23.7	30.65	32.13
pX	33.82	47.29	64.94	76.19	68.05	67.51
u	0.5	0.03	0.43	0.05	0.03	0.06

4.3 Traffic Analysis

In this study the analysis framework focuses on the fundamental differences between the P2P traffic and other Internet traffic (this will be referred to as *non-P2P traffic*). The comparison is done regarding several aspects of the traffic characterization.

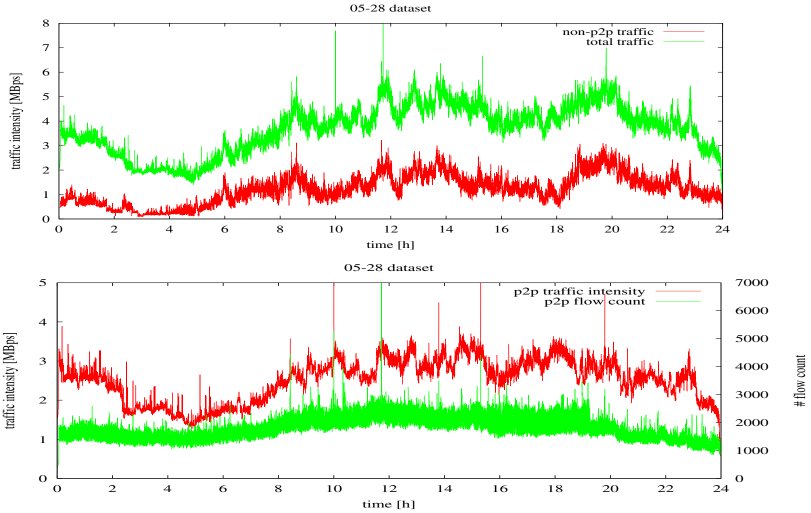


Fig. 2. Traffic intensities from 05_28 data set

Overview of the traffic. The daily fluctuation of the traffic is presented in Fig. 2. The upper plot shows the total and non-P2P traffic intensities of the 05_28 data set while the lower one is the intensity and the flow count of the P2P traffic of the same set.

As observed in general, daily traffic can be divided into two parts: the busy period from around 8h to 24h and the non-busy period from about 0h to 8h. Both P2P and non-P2P traffic follow this daily tendency. However, in the case of non-P2P applications the traffic level shift between busy and non-busy periods is significant (the bandwidth falls to very low values in non-busy period) while this ratio for P2P applications is only around 1/3. This is reasonable since non-P2P users, in general, do not generate traffic in the sleeping time. In contrast, P2P users (in our case also home users) turn on the P2P application and request some audio and video files (some can be very large). Then they leave the system to work over days, even when they are asleep during the night period. Basically, the P2P traffic can be steady over time, which can be seen in Fig. 2: the number of P2P flows has small variation (see the lower plot). We still see a certain decrease in the traffic. It happens since the number of downloadable sources decrease and probably more requests are not added during the night period.

The volume of P2P traffic, see also Table 2, which is about 65% of the total traffic, exceeds by far the traffic volume of the non-P2P applications.

Number of P2P and total active users. In the measurement environment, Internet subscribes do not have fixed IP addresses. Each time a user connects to the Internet, a dynamic address is given to the user. Therefore it is impossible to determine exactly which data flow belongs to which user. However, less error is expected when we choose to associate an individual IP address to a user. Since the ADSL contracts at the present Internet provider do not limit the time

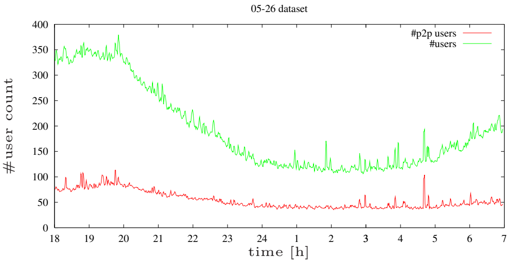


Fig. 3. The average number of (P2P) users (05_26 data set)

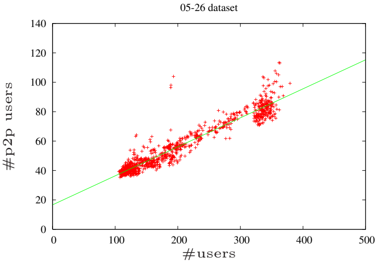


Fig. 4. Relation between P2P users and the total user number (05_26 data set)

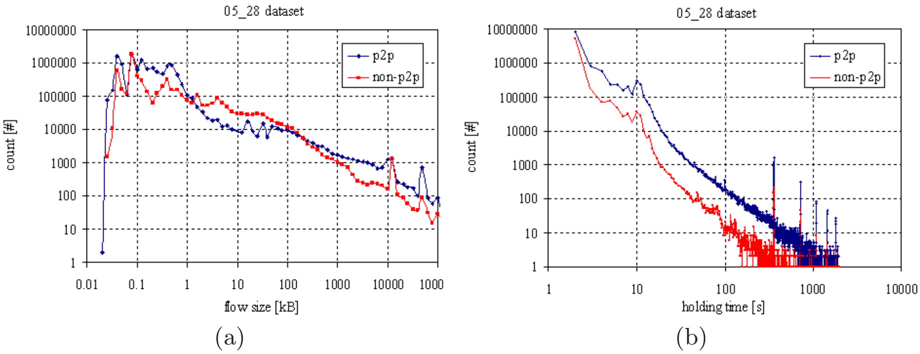
of connections, the average connection time is relatively long. We assume that during our measurements, which lasted at most 24h, only a minimum number of IP address wanderings occurred.

To calculate the number of active users, the number of different IP addresses participating in the flows is counted in every second. Then a sliding window of size 120s and step 50s is applied to smooth the variations caused by communication breaks. One of the results is shown in Fig. 3. The total number of users, according to the time shift between busy and non-busy periods, decays as the non-busy period is approached. The lowest number of users is observed in the non-busy period. This similarity is not so striking in the case of P2P users. The answer is similar to the above, it is due to the typical behavior of P2P users/applications.

The relation between the active P2P users and the total active users is presented in Fig. 4. As seen in the figure, 05_26 data set for example, there is a strong linear connection between the two measures. This means that approximately a fixed ratio of active users is using P2P applications. This is quite an interesting finding and it is hard to find a reasonable explanation. However, if this relation is general, it would be very useful for e.g. traffic dimensioning. We plan to verify this relation in more different network environments. The estimated ratio between P2P users and total users is about 0.2 for this data set, 0.3 for the other two sets.

The relation between the number of active (P2P) users and the occupied bandwidth is also investigated. It is shown that a linear connection can be observed in both cases (P2P and non-P2P traffic). However, the variance of data around the assumed linear function is much higher than in the previous case. In addition, variation is higher and the slope of the line is much lower for non-P2P traffic. The same number of non-P2P users occupy much lower bandwidth compared to that of P2P users.

Flow sizes and holding times. The next comparison is about the properties of data transferring: flow size and flow holding time. Fig. 5(a) presents the histogram of the flow sizes of P2P and non-P2P applications. We find no significant



(a) (b)

Fig. 5. Histogram of flow sizes (a) and flow holding times (b) (05_28 data set)

divergence in this characteristics. In both cases the plots, disregarding flow sizes smaller than 0.1 kB, nearly follow a straight line in the log-log scale. This indicates a possible heavy-tailed (Pareto) model for the flow size for both P2P (with shape parameter a=-0.3) and non-P2P flows (a=-0.25) and also for the overall traffic. (The assumptions of Pareto distribution were verified by several heavy-tailed tests: De Haan's moment method, Hill estimator, and QQ-plot [27].) The number of P2P flows which are larger than about 100 kB is somewhat higher than the number of non-P2P ones, which is also reasonable, but the difference is not significant.

The result seems to be reconcilable with some newer developments of many P2P protocols. Independently of the size of the requested objects, at the beginning the P2P application downloads only a small chunk of the object. The condition of the network and source capacity is predicted from the characteristics of the previous downloads. The size of the next chunk will be determined according to the assumed download quality. Thus, at the end, the P2P traffic (concerning flow size in this case) behaves similarly as the non-P2P traffic.

Similarity is also obtained in the flow holding time distribution of P2P and non-P2P traffic, see Fig. 5(b). Again, in the log-log scale, one can see two almost parallel lines in the two histograms. The plots suggest the Pareto distribution for both cases with the same shape parameter a=1.4. The shift in the histogram plot agrees with the fact that the total number of P2P flows is higher than that of the non-P2P ones by one order of magnitude.

Popularity distribution. The IP addresses were ranked according to their total amount of downloaded traffic. The downloaded traffic is plotted against the ranked IP address (which we have assumed to be associated with an individual user) in Fig. 6. The skewness in the popularity distribution of P2P systems is also justified in our analysis as in many studies of P2P traffic. The top 10% of P2P users corresponds to more than 90% of total download traffic. Our interest, however, is how it differs from the other Internet traffic. Our analysis shows that the difference does not lie at the head of the rank but at the tail. As we go down the rank, the download traffic by ranked users decreases very fast in the case

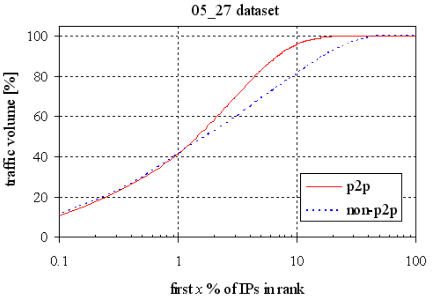


Fig. 6. Traffic volume of ranked IPs (05_27 data set)

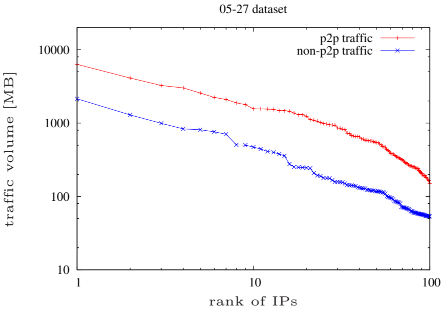


Fig. 7. Traffic vs. top ranked IPs of 05_27 data set

of P2P users. There is a big split between "obsessive" and hobby P2P users. In contrast, the degree of traffic volume decay in case of ranked non-P2P users is very slow. The average non-P2P users create relatively stable traffic when they access the Internet: reading daily news, chatting with friends, etc.

At the top (about 10%) of the ranked list the popular Zipf's law seems to be accurate to describe both P2P and non-P2P traffic popularity. As seen in Fig. 7 two almost linear plot of P2P (marked by +, upper curve) and non-P2P IP rank (marked by x, lower curve) with an approximate slope of -1 indicates the standard Zipf distribution as the suitable model for *top ranked* users' traffic.

Analysis was also carried out for the connection population and similar curves were shown in the results. Fast decrease was observed in the case of P2P traffic as the ranking place increases, the decay is much lower in non-P2P case. In average a normal non-P2P user creates more, and probably smaller connections than P2P users despite that P2P traffic dominates in all measurements both in the volume and the connection number. This happens because, for example, opening of a web page involves multiple downloads of text, many images, and even audio and video elements.

5 Conclusion

In this paper we first presented a novel P2P traffic identification method. The method collects a set of rules derived from the general behavior of P2P traffic. Our method does not use any payload information so it is easy to implement and use when payload cannot be evaluated because of legal or privacy obstacles or cannot be measured due to technical or financial problems. Our validation results show that the proposed algorithm is able to capture the P2P traffic very efficiently. The identification method was used to identify P2P traffic in current measurement data taken from one of the largest Internet providers in Hungary.

We also presented a comprehensive traffic analysis of P2P and non-P2P traffic. The obtained results have highlighted some critical findings. P2P users/applications, by the typical content-sharing objectives of P2P usage, behave in a different way than other Internet applications. The difference manifests itself in the almost stable P2P activities over busy and non-busy time periods, the bandwidth-hungry nature, the skewness in the traffic volume distribution between P2P users, etc. However, the characteristics of P2P traffic aggregation, which would be a more important aspect from the service providers' and network operators' point of view, are quite similar to those of other traffic aggregation. While in the beginning P2P applications were confined to greedy file-sharing, nowadays they have grown up to be an unisolable component of the Internet due to several refined developments of P2P protocols. It has been shown that there is always a certain ratio of home users who use some P2P applications. The study establishes that the workload of P2P applications generates similar (heavy-tailed) flow size and flow holding time distribution like several non-P2P applications. As a consequence the P2P aggregation also shows a similar characteristics.

There may come the time when we should change the way of thinking about and treating P2P traffic. It is not an outstanding but an inseparable part of the overall Internet traffic just like every other traffic component. Our future work will focus on the research to further investigate this conjecture for general Internet traffic.

References

1. S. Saroiu, K. P. Gummadi, R. Dunn, S. D. Gribble, H. M. Levy, "An Analysis of Internet Content Delivery Systems", in Proc. 5th Symposium on Operating Systems Design and Implementation, Boston, MA, USA, Dec. 2002.
2. S. Sen, J. Wang, "Analyzing Peer-to-Peer Traffic Across Large Networks", *IEEE/ACM Transactions on Networking*, 12(2):219-232, 2004.
3. K. Tutschku, "A Measurement-based Traffic Profile of the eDonkey Filesharing Service", PAM 2004: 12-21.
4. J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips, "The Bittorrent P2p File-Sharing System: Measurements And Analysis", 4th Int. workshop on Peer-to-Peer Systems (IPTPS'05), Feb. 2005.
5. Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, D. Towsley, "Modeling Peer-Peer File Sharing Systems", in Proc. INFOCOM'03, San Francisco, CA, Mar. 2003.
6. K. K. Ramachandran, B. Sikdar, "An Analytic Framework for Modeling Peer to Peer Networks", in Proc. INFOCOM'05, 2005.
7. G. de Veciana, X. Yang, "Fairness, Incentives and Performance in Peer-to-Peer Petworks", in Proc. Allerton Conf. on Communication, Control and Computing, 2003.
8. X. Yang, G. de Veciana, "Service Capacity of Peer to Peer Networks", in Proc. INFOCOM'04, 2004.
9. D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", in Proc. ACM SIGCOMM'04, Portland, OR, Aug. 2004.
10. B. Yang, S. Kamvar, H. Garcia-Molina, "Addressing the Non-Cooperation Problem in Competitive P2P Systems", Workshop on Peer-to-Peer and Economics, Jun. 2003.

11. D. Hughes, I. Warren, G. Coulson, “Improving QoS for Peer-to-Peer Applications through Adaptation”, in Proc. of the 10th Int. Workshop on Future Trends in Distributed Computing Systems (FTDCS 2004), Suzhou, China, May 26-28, 2004.
12. E. Kalyvianaki, I. Pratt, “Building Adaptive Peer-To-Peer Systems”, in Proc. 4th Int. Conf. on Peer-to-Peer Computing (P2P’04), 2004.
13. M. Iguchi, M. Terada, K. Fujimura, “Managing Resource and Servent Reputation in P2P Networks”, in Proc. 37th Annual Hawaii Int. Conf. on System Sciences (HICSS’04), 2004.
14. G. Ding, B. Bhargava, “Peer-to-Peer File-Sharing over Mobile Ad hoc Networks”, in Proc. PerCom Workshops, 2004.
15. M. Demirbas, H. Ferhatosmanoglu, “Peer-to-Peer Spatial Queries in Sensor Networks”, in 3rd IEEE Int. Conf. on Peer-to-Peer Computing (P2P’03), Linkoping, Sweden, Sept. 2003.
16. M. Roussopoulos, M. Baker, D. S. H. Rosenthal, T. J. Giuli, P. Maniatis, J. C. Mogul, “2 P2P or Not 2 P2P?”, IPTPS 2004: 33-43.
17. Y. Guo, K. Suh, J. Kurose, D. Towsley, “A Peer-to-Peer On-Demand Streaming Service and Its Performance Evaluation”, in Proc. IEEE Int. Conf. on Multimedia & Expo (ICME 2003), Baltimore, MD, Jul. 2003.
18. G. Cugola, G. P. Picco, “Peer-to-Peer for Collaborative Applications”, Int. Workshop on Mobile Teamwork Support, Vienna, Austria, Jul. 2002.
19. C. Buragohain, D. Agrawal, S. Suri, “A Game Theoretic Framework for Incentives in P2P Systems”, in Proc. 3rd Int. Conf. on Peer-to-Peer Computing, 2003.
20. T. Risse, P. Knezevic, A. Wombacher, “P2P Evolution: From File-sharing to Decentralized Workflows”, *Information Technology*, 4:193–199, Oldenbourg, 2004.
21. S. Sen, O. Spatscheck, D. Wang, “Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures”, in Proc. 13th Int. Conf. on World Wide Web, NY, USA, 2004.
22. M. Kim, H. Kang, J. W. Hong, “Towards Peer-to-Peer Traffic Analysis Using Flows”, DSOM 2003: 55-67.
23. T. Karagiannis, A. Broido, M. Faloutsos, K. Claffy, “Transport Layer Identification of P2P Traffic”, in Proc. 4th ACM SIGCOMM Conf. on Internet Measurement, Taormina, Sicily, Italy, Oct. 25-27, 2004.
24. Internet2 NetFlow: Weekly Reports - http://netflow.internet2.edu/weekly/
25. A. Gerber, J. Houle, H. Nguyen, M. Roughan, S. Sen, “P2P The Gorilla in the Cable”, in National Cable & Telecommunications Association (NCTA) 2003 National Show, Chicago, IL, June 8-11, 2003.
26. S. Ohzahata, Y. Hagiwara, M. Terada, K. Kawashima, “A Traffic Identification Method and Evaluations for a Pure P2P Application”, Lecture Notes in Computer Science, p55 Vol. 3431, 2005.
27. S. I. Resnick, “Heavy Tail Modeling and Teletraffic Data”, *The Annals of Statistics*, 25(5):1805–1869, 1997.

A Decentralized Recommendation System Based on Self-organizing Partnerships

Giancarlo Ruffo, Rossano Schifanella, and Enrico Ghiringhello*

Dipartimento di Informatica, Università di Torino,
Corso Svizzera, 185 - 10149, Torino (Italy)
{ruffo, schifane}@di.unito.it

Abstract. Small World patterns have been found in many social and natural networks, and even in Peer-to-Peer topologies. In this paper, we analyze File Sharing applications that aggregate virtual communities of users exchanging data. In these domains, it is possible to define overlaying structures that we call "Preference Networks" that show self organized interest-based clusters. The relevance of this finding is augmented with the introduction of a proactive recommendation scheme that exploits this natural feature. The intuition behind this scheme is that a user would trust her network of "elective affinities" more than anonymous and generic suggestions made by impersonal entities.

Keywords: Peer-to-Peer, Recommendation Management, Small World Networks, Social Networks.

1 Introduction

Even if File-Sharing is not the only application of the peer-to-peer paradigm, it is indeed a unique environment where (a subset of) social attitudes of p2p users can be studied and deeply observed. This is mainly due to the huge popularity of tools like Gnutella, eMule, bitTorrent, and so on, that every day attract millions of users that share several terabytes of electronic information. Even if maybe nobody is able to extract generalizable applicative consequences from phenomena observed in a particular *ecosystem*, it is indeed true that the in depth understanding of the way users strictly cooperate for reaching their individual aims has a significant scientific relevance. Furthermore, the new knowledge about a given community, that uses a particular kind of application, can really be exploited for improving the application itself, and even for improving other applications based on the same paradigm.

Many file sharing users know very well how much is difficult to find something interesting just picking up, at random, another user's file list. This is mainly due to the fact that every person has different likings, and an item, interesting for one user, can be absolutely detestable for someone else. Conversely, when the search process is iterated on different queries in an unstructured network (e.g., Gnutella, FastTrack, and so on), it is quite surprisingly to observe that there is a constant core in the set of responding users. In fact, many file sharing clients allow the participant of a network to create lists

* E. Ghiringhello joined this project during the preparation of his bachelor thesis.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 618–629, 2006.

of *favourite users*, in order to directly explore the given shared file systems that are more likely to return something interesting to the querying user.

During the years, many social networks have been discovered and analysed, e.g., the scientific co-authorship network [1], the friend-ship network [2], and so on. In this paper, we want to analyse a particular instance of the *Preference Network*, that links users sharing common interests. In other words, we create a (family of) network(s) whose nodes are users of a file sharing system, and whose links connect pair of nodes that share one or more identical files. The main purpose of this study is to empirically prove that a Preference Network has a *Small World* topology [3, 4]. As a relevant consequence, we want to propose an applicative framework, where the small world property of the preference graph in unstructured p2p networks, can be exploited to return a decentralized recommendation service to the user.

2 Related Work and Road Map

This paper contains two contributions: (1) we introduce a family of interest based graphs on top of Gnutella, which connect users sharing at least m common files, and proving that they show small world topologies; (2) a practical recommendation scheme is proposed to the file sharing community, that takes advantage of the high clustering coefficient of the previously introduced Preference Networks.

The definition of preference networks is somehow similar to the one given for *data-sharing graphs* in [5, 6], with a subtle, but decisive, difference: two users are connected in a preference network when they are storing (and sharing) replica of at least m common files - uniquely identified by means of a hash code. Conversely, two users are connected in a data-sharing graph if they (try to) download at least m common objects during a time interval T - identified by way of their file names. Hence, the two structures differ (1) at scope level, (2) at data collection level, and (3) at temporal level. First of all, we are interested in the tastes of the users, and we want to connect people that have similar likings. Secondly, Leibowitz and al., as reported in [5], collected data from processing HTTP logs at a large Israeli ISP. They focused on traffic generated by KaZaa clients, and they refer to file names to find out if different users were downloading the same items. However, this does not capture some phenomena that are relevant in file sharing systems, such as the presence of fake files (i.e., items having names not matching their contents), the rapid downloading-deleting process (i.e., very often a user downloads a file and suddenly she deletes it because she realizes that it is out of interest), and the possibility for a file to have many replica with different names. For these reasons, we preferred to collect *QueryHit* [7] messages, stored by running for several days a (modified) Gnutella Ultra-Peer: these messages contain the precise information of (some of) the files actually shared by the network (with the hashed file identifier, too). Finally, we deliberately did not consider any temporal constraints, because we want to study persistent phenomena. In fact, we are making the assumption that if a user is still sharing a file, then she directly inserted it in the network, otherwise she previously downloaded it. In the first case, the user is trivially interested in that kind of content. In the second case, if the user downloaded it and she did not delete the file immediately after, we can reasonably assume that she is interested in it. Observe that we are consciously ignoring transient events featuring in data sharing, because in this phase of the study, we are not interested in how the snapshots of the network change in different instances of time: if a user is found showing a preference for a given item, then he will maintain an interest for it (e.g., if he likes the Beatles classic ”Yesterday”, then he will very likely love that song even in the future). On the contrary, the domain investigated in [5] is highly dynamic, and it is subject to change very rapidly. If the reader is interested to characterize dynamic phenomena in unstructured topologies, then she needs a parallel analysis, as explained in [8].

Preference Networks are, indeed, very similar to the structures described in [9]. In that study, the inefficient Gnutella search mechanism based on flooding is enhanced by means of interest based localities. Our paper adds two contributions to that work: first of all we experimentally prove that preference networks are small worlds. This result is generalizable outside Gnutella, because it is not related to the file sharing network topology or to the given search mechanism. Secondly, we propose a recommendation scheme that suggest users the next likely interesting files, without concerning about the localization of the item, because this can be performed with many known efficient solutions (e.g., by way of a Distributed Hash Table based algorithm).

Recommendations and Trust Management are fertile areas in the Peer-to-Peer scientific community [10]. Differently to previous work in this field, our recommendation scheme acts *proactively* pushing automatic suggestions to the user, presenting her an *unseen* file. This point is somehow related to many e-commerce services, that assist the user with sentences like “Customers who bought this CD also bought: The Rolling Stones - Aftermath”. In fact, to our knowledge, this is the first proactive recommendation proposal that works in a complete distributed domain without any central repository nor any common background knowledge. Suggestions are made uniquely on the basis of natural and self organizing users’ partnerships. Like in the real world, even in virtual social communities, people can meet others with common interests and trust them by word of mouth before buying or looking for items.

Sections 3 and 4 will be devoted to define preference networks and to provide an empirical proof of the small world features of such graphs. The recommendation scheme is introduced in Section 5. Section 6 reports some conclusion and the agenda of our ongoing work.

3 Graphs, Topologies and Preference Networks

Let $G = (V, E)$ be a graph, where V and E are respectively the set of vertices and the set of edges between nodes. Let L be the *average shortest path length*, and let C be the *clustering coefficient* of G. The clustering coefficient of a graph gives a measure of how many almost complete sub-graphs are in the topology. In fact, given $v_i \in V$, the clustering coefficient C_i is the ratio of the actual number of edges between neighbors of v_i for the maximum value of such a number. The clustering coefficient C of the graph is the mean value of all the C_is. Formally speaking, if we define the set of *neighbors* of v_i as $V_i = \{v_j\} : v_j \in V, e_{ij} \in E$, then the *degree* of v_i is $d_i = |V_i|$, i.e., d_i is the number of neighbors of the vertex. Note that D_i, the maximum number of links between neighbors of v_i, can be defined in function of d_i; in fact, if G is a directed graph (i.e.,$e_{ij} \neq e_{ji}$), then $D_i = d_i \cdot (d_i - 1)$. Otherwise (when G is undirected), $D_i = \frac{d_i \cdot (d_i - 1)}{2}$. Let $E_i = \{e_{jk}\} : v_j, v_k \in V_i, e_{jk} \in E$ be the actual set of edges between neighbors of v_i. Hence, the *clustering coefficient* of v_i can be defined as:

$$C_i = \frac{|E_i|}{D_i}.$$

Observe that if C_i is equal to 0, it means that the neighbors of v_i are not connected each other (i.e., $E_i = \emptyset$).

Otherwise, if $C_i = 1$, then the sub-graph G_i is complete, where $G_i = (V_i \cup \{v_i\}, E_i \cup \{e_{ij} : e_{ij} \in E\})$.

Furthermore, the *clustering coefficient of graph G* is defined as in [3]:

$$C = \frac{\sum_i C_i}{|V|}.$$

Even if Newmann [1] describes in a different manner the clustering coefficient, we recall that both definitions bring to comparable results. Therefore, in the following lines the reader should remember that the previous definitions were used during our analysis, and that a different clustering definition would not affect the given conclusions.

We can intuitively think at a small world graph, as a loosely connected network of (almost) complete sub-graphs. Hence, a graph G is checked against this property by comparison with a random graph G_{rand} with the same number of vertices and edges. Let L_{rand} and C_{rand} be respectively the average shortest path length and the clustering coefficient in the random graph, we say that G is *small world* if $C \gg C_{rand}$ and $L \approx L_{rand}$.

In order to further model our domain, let us assume that a set of users $U = \{u_1, u_2, \ldots, u_n\}$[1] is sharing a set of items $S = \{s_1, s_2, \ldots, s_l\}$. We map users to items with function $f : U \to \mathcal{P}(S)$, where $\mathcal{P}(S)$ is the power set of S. Of course, $\bigcup_{i=1}^{n} f(u_i) = S$. Moreover, in our environment, we assume that for some i and j, $f(u_i) \cap f(u_j) \neq \emptyset$, i.e., users may share (some) identical files. Observe, that this assumption is realistic in an unstructured overlay network, where *users share what they want to*.

These hypotheses enable us to introduce the concept of a *preference network*, that we can model with a graph where each vertex corresponds to a different user, and a link is connected between two users that share at least m items; i.e., $G^m = (U, E^m)$, where $e_{ij}^m \in E^m \Leftrightarrow |f(u_i) \cap f(u_j)| \geq m$.

In next section, we want to show that there is a natural partnership between users. In particular, we found that preference networks with $2 \leq m \leq 8$ have a small world topology, meaning that we can assume a transitivity property between users: let u_a and u_b be two users sharing at least m common files, and let u_c be a user that shares at least m files with u_b, then it is very likely that u_a and u_c share at least m files. This would be just a corollary of the small world property, because if G^m is small world, than it will have a clustering coefficient C with a high value. This *triangulation* between users is very relevant to our study, because these self-organizing communities

[1] In the rest of the paper, we will assume a bijection between users and nodes of the p2p file sharing network. Hence, we can use u_i to indicate the i-th node as well as the i-th user.

can be exploited to a very efficient and fully decentralized recommendation scheme, as described in Section 5.

4 Data Collection

In order to study the small-world properties of the so defined *preference networks*, we perform the following preliminary steps: (1) implementation of a Gnutella network crawler, (2) data collection, (3) post-processing of the gathered traces, and (4) generation of the preference graphs.

As described above, we focus on persistent phenomena of data-sharing relationships between users, hence we are interested in tracing *QueryHit* messages exchanged between peers. The modern Gnutella network topology consists in a *two-tier* overlay where a set of interconnected *ultrapeers* forms the top-level overlay to which a large group of *leaves* are connected. Leaves never forward messages: they send queries to the ultrapeers and wait for a set of *QueryHit*s matching the searching criteria. Otherwise, an ultrapeer acts as a *proxy* to the Gnutella network for the leaves connected to it. Ultrapeers are connected to each other and to *regular* Gnutella hosts. *QueryHit* messages return back to the querying user by reverse path forwarding. This ensures that only those servents that routed the *Query* message will get the returning *QueryHit* message. Therefore, a *ultrapeer* receives all *QueryHit* messages addressed to its leaves.

Table 1. Data collected by the Gnutella ultrapeer crawler from 19 October to 26 October 2005

CHARACTERISTICS OF TRACES COLLECTED

Time Interval	7 days
# Users (distinct IP)	136.752
# Files (distinct SHA)	473.848
# Query Hits	1.601.610
# Query Hits (mp3 songs)	798.821

To hit the mark, we have modified the Gnutella servent *Phex* [11][2], an open-source client written in Java language. Our crawler is forced to access the network in the *ultrapeer* mode, and to trace down all the *QueryHit* messages it stores and forwards. Collected data contains information about the user that answers the query and the related resources he shares. Each user is identified by the IP address, whereas the hash code of the resource is exploited to unambiguously identify the file. The crawler collects a seven days of Gnutella traffic (from 19 October to 26 October 2005). As described in Table 1, the traces are composed by more than 1.5 millions data entries generated by about 130.000 distinct IP addresses and involving 473.848 different SHA-1 file hashes. Notice that each *QueryHit* message can contain more than one reference to resources matching the search criteria[3]. Figure 1 shows the file popularity distribution observed in our Gnutella snapshot. Let us notice that it follows Zipf's law, as already observed in [6].

[2] We used the Phex version 2.6.4.89.

[3] We found that all the received *QueryHit* messages contained at most five query results.

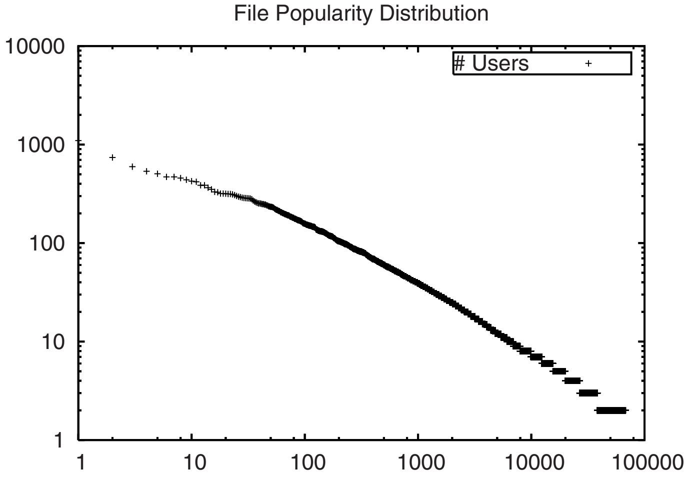


Fig. 1. File popularity distribution, plotted in a log-log scale, follows Zipf's law

After the raw data is collected, we apply a filtering phase composed by the following steps:

- *Private network IP address filtering:* our model binds each IP address to a distinct user. Indeed, it is possible that the same IP address corresponds in fact to different users, e.g. shared workstations or presence of NAT/proxy. A private network environment provides a concrete example of this effect: let us suppose that the IP 192.168.1.10 publishes a set of resources R. We could relate the IP 192.168.1.10 to a particular user u and we could wrongly assert that u shares the files belonging to R. In fact, many distinct users in different networks can obtain this address, so that the *QueryHit* content cannot distinguish between these users. The effect is the presence of distinct IPs that seem to share large sets of files, affecting the fairness of the preference graphs [4]. To get rid of this effect, we filter out all IP addresses that belong to the private network class specification[5].

  Notice that the opposite phenomenon can be observed as well. For example, in a DHCP-based network the same user can obtain different IP addresses in distinct sessions. Therefore, a set of resources R that effectively belong to a user u, can be seen as sum of shared items from many users. Obviously, this phenomenon can smooth the hub behavior of the user u. However, we think that this effect does not impact our study due to the relatively short time of trace collection[6].

[4] Indeed, these IPs behave like hubs, so they should amplify the small-world properties showed by the preference graphs.

[5] We filter out the following sets of IP addresses[12]: $10.x.x.x$, $192.168.x.x$ and the range from 172.16.0.0 to 172.31.255.255.

[6] We executed different monitoring sessions using the Gnutella servent's IDs to discriminate between users. These IDs are generated by running a cryptographic hash function on a random input value, so that it changes after each login. We did not observe relevant differences in the results w.r.t. to the ones obtained from the filtered data, thus enforcing our findings.

Table 2. *Average shortest path length* L and *clustering coefficient* C for the preference networks

G^m	# Nodes	# Edges	Preference Graph L	Preference Graph C	Random Graph L_{rand}	Random Graph C_{rand}
G^2	22777	428931	3.29	0.43	3.418	0.0017
G^3	9807	81088	3.53	0.37	4.351	0.0017
G^4	4779	23378	3.68	0.35	5.336	0.0020
G^5	2612	8519	3.81	0.33	6.655	0.0025
G^6	1501	3617	3.93	0.28	8.316	0.0032
G^7	891	1780	4.12	0.27	9.815	0.0045
G^8	591	990	4.4	0.25	12.371	0.0057

- *Focusing on MP3 songs:* in our evaluation we consider only entities related to mp3 song files. Filtering out other content types allows a better analysis about user preferences relationships in a specific field, e.g. the music likings domain. Moreover, the cleaned dataset reduces the complexity inherent to the graphs generation task. However, Table 1 shows that about 50% of the overall resources were mp3 songs.

After the filtering step, we generated the preference graphs G^m, where nodes are users and a link connects two users that share at least m items (see Section 3). We created several graphs, from G^2 to G^8, and for each of them we computed the average shortest path length (L) and the clustering coefficient (C). These metrics are estimated also for a random graph with identical number of nodes and edges (L_{rand} and C_{rand}). As Table 2 shows, all G^m graphs reveal small-world patterns: in fact, we have that, for all the preference networks, $C \gg C_{rand}$ and $L \approx L_{rand}$ (very interestingly, it always happens that $L < L_{rand}$).

Table 3. Example of C and L for several real networks

$Network$	L	C	$Reference$
WWW, site level, undir.	3.1	0.1078	Adamic, 1999
Movie actors	3.65	0.79	Watts and Strogatz, 1998
LANL co-authorship	5.9	0.43	Newman, 2001a, 2001b, 2001c
MEDLINE co-authorship	4.6	0.0666	Newman, 2001a, 2001b, 2001c
SPIRES co-authorship	4.0	0.726	Newman, 2001a, 2001b, 2001c
NCSTRL co-authorship	9.7	0.496	Newman, 2001a, 2001b, 2001c
Math. co-authorship	9.5	0.59	Barábasi et al., 2001
Neurosci. co-authorship	6	0.76	Barábasi et al., 2001
E. coli, substrate graph	2.9	0.32	Wagner and Fell, 2000
E. coli, reaction graph	2.62	0.59	Wagner and Fell, 2000
Ythan estuary food web	2.43	0.22	Montoya and Sole’ , 2000
Silwood Park food web	3.40	0.15	Montoya and Sole’ , 2000
Words, co-occurrence	2.67	0.437	Ferrer i Cancho and Sole , 2001
Words, synonyms	4.5	0.7	Yook et al., 2001b
Power grid	18.7	0.08	Watts and Strogatz, 1998
C. Elegans	2.65	0.28	Watts and Strogatz, 1998

Notice that the data-sharing relationships collected by means of tracing *QueryHit* messages represents obviously only a fraction of the resources shared within the network. Therefore we reasonably think that a global vision could strongly confirm and enhance the small-world properties observed.

In Table 3 we compare the values of L and C with other known domains showing small-world phenomenons [13].

5 A Recommendation Scheme Based on Preference Partnership

Users' preferences can be used for improving search mechanisms in overlay network, as suggested in [5], to enforce ontologies definitions and routing in semantic (p2p) networks [14, 15], and also to locate content avoiding expensive flooding search mechanisms [9]. We are interested to the highly informative power of self-organizing communities characterized by (almost) complete sub-graphs: users in the same cluster share each-other a subset of common items and are likely interested to other files popular in the cluster. The transitivity property may be used for enabling *reserved information lanes* between users, in order to announce items that are potentially of interests for members of the same cluster.

First of all, let us introduce a notation that we will use in the rest of the section. Given $u_x, u_y \in U$, and an item $s_k \in S$, we describes the event that u_x downloaded file s_k from u_y (or, similarly, u_y uploaded s_k to u_x) with $u_y \overset{s_k}{\rightarrow} u_x$. Of course, if $u_y \overset{s_k}{\rightarrow} u_x$, then $s_k \in f(u_x) \cap f(u_y)$ (even if the adverse implication is not always applicable).

For each user $u_i \in U$, we define:

$$F_0(u_i) = \{u_j : (i \neq j) \wedge \exists s_k (u_i \overset{s_k}{\rightarrow} u_j \vee u_j \overset{s_k}{\rightarrow} u_i)\},$$

that is the *set of contacts* of u_i. Roughly speaking, the node of user u_i maintains a list of other users that exchanged some files with u_i. In order to exploit the triangulation property of the preference networks which the node is connected to, we consider also the *set of contacts of the first order* of u_i:

$$F_1(u_i) = \bigcup_{u_j \in F_0(u_i)} F_0(u_j).$$

We introduce the *list of friends (or partners)* of u_i as it follows:

$$F(u_i) = F_0(u_i) \cap F_1(u_i).$$

Note that relationships in $F(u_i)$ are stronger than in $F_0(u_i)$ (see Figure 2).

The node u_i stores an integer value $m(u_i, u_j)$ for each reference in $F(u_i)$. More precisely, we define the *partnership degree* of the pair u_i and u_j as $m : U^2 \rightarrow \mathcal{N}^+$, where $m(u_i, u_j) = |f(u_i) \cap f(u_j)|$, that is the number of files that they have in common. For the sake of simplicity, in the rest of the section, we will use the notation m_{ij} instead of $m(u_i, u_j)$.

At an implementation level, list $F(u_i)$ has a constant size, and it is ordered on the basis of the value of the partnership degree m_{ij}: the user on the top of the list has more

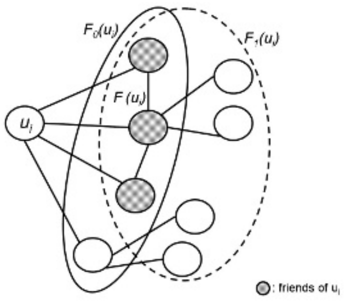


Fig. 2. Example: contacts and friends of u_i

files in common with u_i than with the others. On the contrary, the less "interesting" user (e.g., $m_{ij} \approx 0$), is likely to be removed from the list. The reader should observe that, given $\overline{m}$, it is possible to extract, from this list, the (known) neighbors of u_i in the preference graph $G^{\overline{m}}$; in fact, it is easy to note that $m(u_i, u_j) = \overline{m} \Rightarrow u_j \in U_i^{\overline{m}}$, where $U_i^{\overline{m}}$ is the set of neighbors of u_i in $G^{\overline{m}}$.

For example, let us suppose that user u_x downloaded s_1 and s_2 from u_y. Moreover, he downloaded s_3, s_4 and s_5 from u_z, and s_6 from u_v. Finally, we have also that $F_0(u_x) = F_1(u_x)$. As a consequence, we have that: $f(u_x) = \{s_1, s_2, s_3, s_4, s_5, s_6\}$, and $F(u_x) = \{u_y, u_z, u_v\}$. Furthermore, after having interacted with u_y, u_z and u_v, the p2p client of u_x got also their file lists. So, u_x knows that $f(u_y) = \{s_1, s_2, s_4, s_7\}$, $f(u_z) = \{s_3, s_4, s_5, s_7, s_8\}$ and $f(u_v) = \{s_6, s_7, s_3, s_4, s_5\}$. The values of function m are updated after each interaction. After the last download, they are as it follows: $m_{xy} = 3, m_{xz} = 3$, and $m_{xv} = 4$. $F(u_x)$ is ordered as it follows: (u_v, u_y, u_z).

We need also to identify the set of files[7] owned by the friends of u_i, but that are not possessed by u_i:

$$\text{Co-f}(u_i) = (\bigcup_{u_j \in F(u_i)} f(u_j)) - f(u_i).$$

The state of a running node includes also a *file map*, that returns the partners owning a given resource, that is not possessed by u_i. Hence, we define the family of functions:

$$map_i : \text{Co-f}(u_i) \rightarrow \mathcal{P}(F(u_i)),$$

where $map_i(s_k) = \{u_j \in F(u_i) : s_k \in \text{Co-f}(u_i) \cap f(u_j)\}$.

In the previous example, we have that $\text{Co-f}(u_i) = \{s_7, s_8\}$, $map_i(s_7) = \{u_y, u_z, u_v\}$, and $map_i(s_8) = \{u_z\}$.

5.1 *Die Wahlverwandtwchaften*: The Intuition

The intuition behind the proposed recommendation scheme is based on the observation that friends of a given peer build a cluster of nodes with different partnership degrees.

[7] Note that the p2p client of u_i does not store all the friends' files, but only an unique reference to them (e.g., their SHA-1 hashes).

We previously observed in Section 4 that nodes can be naturally gathered together on the basis of common interests. Moreover, we noted that there are peers that are more kindred to some partners than others; in fact, we found that a preference network G^m is a small world, even with growing values of m. But not all the nodes involved in preference networks with lower degrees than m are still involved in G^m. Thus, some relationship between nodes in the same cluster is stronger than others: even in the file sharing community, *elective affinities* [16] rule the social behavior of the users.

We want to sort files in Co-f(u_i) by means of the following criteria:

1. *popularity* in the cluster of partners of u_i;
2. *partnership degree* of friends storing the missing files;

The *recommendation list* is defined as the ordered sequence:
$R(u_i) = (s_{k_1}, s_{k_2}, \ldots, s_{k_\ell})$, where $\ell = |\text{Co-f}(u_i)|$, and $\forall h = 1, \ldots, \ell : s_{k_h} \in$ Co-f(u_i). Files in $R(u_i)$ are sorted (and, hence, recommended), on the basis of the weight defined below:

$$w(s_{k_h}) = \frac{\sum\limits_{u_j \in map_i(s_{k_h})} (m_{ij})}{max_d(|map_i(s_{k_d})|)},$$

i.e., $\forall s_{k_d} \in R(u_i) : w(s_{k_{d-1}}) \leq w(s_{k_d}) \leq w(s_{k_{d+1}})$.

In our example, files will be recommended to u_i in this order: (s_7, s_8). In fact, we have that $w(s_7) = 3.\bar{3}$, and $w(s_8) = 1.0$. Of course, in a practical environment, we can set a threshold, in order to filter out recommendations with low weight. In the previous case, if such a threshold is set to 2.0, only file s_7 would be submitted to the user's attention. The estimation of this value is one of the tasks of on-going work.

5.2 Discussion

Let us numerically quantify the popularity of a file and the average partnership degree of nodes hosting a given item as it follows:

Given a node u_i, the *popularity* of a missing file s_{k_h} is calculated by way of the family of functions $\text{pop}_i :$ Co-f$(u_i) \rightarrow]0, 1]$, where

$$\text{pop}_i(s_{k_h}) = \frac{|map_i(s_{k_h})|}{max_d(|map_i(s_{k_d})|)}.$$

Given a node u_i, the *degree* of a missing file s_{k_h} as the average partnership degree of nodes in $F(u_i)$ that stores s_{k_h}. This value is calculated by way of the family of functions $\text{deg}_i :$ Co-f$(u_i) \rightarrow \mathcal{R}^+$, where

$$\text{deg}_i(s_{k_h}) = \frac{\sum\limits_{u_j \in map_i(s_{k_h})} (m_{ij})}{|map_i(s_{k_h})|}.$$

Trivially, $w(s_{k_h}) = \text{pop}_i(s_{k_h}) \cdot \text{deg}_i(s_{k_h})$.

The following theorem simply shows that the recommendations are sorted according to the criteria inspired by the preference networks which the node is connected to: a missing file is suggested for its popularity amongst the friends of the users and for *affinity* degree of the node that stores the given file.

Theorem 1. Given a node u_i, and two files s_{k_x} and s_{k_y} in Co-f(u_i), s.t., $w(s_{k_x}) > w(s_{k_y})$, then the following statements are true:

1. If the files have the same popularity, then s_{k_x} is owned mostly by nodes with a higher average partnership degrees w.r.t. s_{k_y}.
2. If the files are owned by nodes with the same average partnership degree, then s_{k_x} is more popular than s_{k_y} in Co-f(u_i).

Proof. Note that the hypothesis says that $w(s_{k_x}) > w(s_{k_y})$, that means that $\text{pop}_i(s_{k_x}) \cdot \text{deg}_i(s_{k_x}) > \text{pop}_i(s_{k_y}) \cdot \text{deg}_i(s_{k_y})$.

It is easy to show that, when $\text{pop}_i(s_{k_x}) = \text{pop}_i(s_{k_y})$ (> 0), then it follows that $\text{deg}_i(s_{k_x}) > \text{deg}_i(s_{k_y})$, which proves the first part of the theorem.

The second enunciation states that, on the contrary, $\text{deg}_i(s_{k_x}) = \text{deg}_i(s_{k_y})$ (> 0); in this case, we have that $\text{pop}_i(s_{k_x}) > \text{pop}_i(s_{k_y})$, which proves the theorem.

5.3 Implementation

For evaluation purposes, we implemented a recommendation module that can be integrated to a previously installed Phex servent. During testing, we noted that only after few interactions, the list of friends of the given node becomes quite long: Phex acts in a multi-download fashion, hence after only one download, we have as many contacts as the number of different parts of the divided file. Indeed, lists of friends and files grow very quickly, and as a consequence, in order to preserve lightness of the peer's state, lists are forced to be limited in size. Optimizations based on Bloom filters are under study. Anyway, the small world property of the preference networks makes the files and the partners lists' lengths to stabilize after a while.

Of course, we need to spread the module to many users, and ask them to allow us to monitor part of their activities (without asking them to tell us what they share, for privacy reasons). There are maybe many different scenarios that cannot be foreseen at the moment, and that only a controlled trial period of the package can reveal.

6 Conclusions and Ongoing Work

We defined the concept of preference networks, and we showed that such graphs, built by means of data collected by a modified Gnutella Ultra-Peer, are characterized by small world topologies. Moreover, starting from these findings, we described a fully decentralized recommendation scheme that can be easily implemented in many popular p2p file sharing clients. The most of the relevance of such a result is that only information given by self organizing communities and natural clusters of partnerships are taken into consideration, without defining any semantic knowledge and any ontology.

We also implemented a open source prototype for the Phex Gnutella servent, that will be used for future analysis and evaluations.

Acknowledgment

Part of this work has been financially supported by the Italian FIRB 2001 project number RBNE01WEJT “Web MiNDS”.

References

1. M.E.J.A. Newman. A study of scientific co-authorship networks. *Journal Physics Review*, 20, 2000.
2. T. J. Fararo and M. Sunshine. *A Study of a Biased Friend-ship Network*. Syracuse University Press, 1964.
3. D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440442, 1998.
4. M. E. J. Newman. Models of the small world. *J. Stat. Phys.*, 101:819–841, 2000.
5. N. Leibowitz, M. Ripeanu, and A. Wierzbicki. Deconstructing the kazaa network. In *Proc. of the Third IEEE Workshop on Internet Applications*. IEEE Press, June 2003.
6. I. Foster A. Iamnitchi, M. Ripeanu. Small-world file-sharing communities. In *The 23rd Conference of the IEEE Communications Society (InfoCom 2004)*, Hong Kong, 2004.
7. Gnutella 0.6 Protocol Specification. http://www.gnutella2.com/index.php/-main_page#the_protocol.
8. D. Stutzbach, R. Rejaie, and S. Sen. Characterizing unstructured overlay topologies in modern p2p file-sharing systems. In *Proc. of the ACM SIGCOMM Internet Measurement Conference*, October 2005.
9. K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-topeer systems. In *InfoCom*, 2003.
10. Girish Suryanarayana and Richard N. Taylor. A survey of trust management and resource discovery technologies in peer-to-peer applications. Technical report, UC Irvine, 2004.
11. Phex Gnutella Client. http://phex.kouk.de/mambo/.
12. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de, and E. Lear. Address allocation for private internets. RFC 1918, Internet Engineering Task Force, February 1996.
13. R. Albert. *Statistical mechanics of complex networks*. PhD thesis, 2001.
14. Crespo and Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Computer Science Department, Stanford University, 2002.
15. N. Borch. Improving semantic routing efficiency. In *Proc. of the 2nd Inter. Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P’05)*, July 2005.
16. J. W. von Goethe. *Die Wahlverwandtschaften*. 1809.

Enhancing the P2P Protocols to Support Advanced Multi-keyword Queries

Samir Ghamri-Doudane[1] and Nazim Agoulmine[2]

[1] LIP6-CNRS, University of Paris 6, France
8, rue du Capitaine Scott – 75015 – Paris, France
samir.ghamri-doudane@lip6.fr
[2] LRSM, University of Evry, France
nazim.agoulmine@lip6.fr

Abstract. Recently, Peer-to-Peer has become a popular paradigm for building distributed systems, aiming to provide resource localization and sharing in large-scale networks. However, advanced searching for resources remains an open issue. The flooding technique used by some Peer-to-Peer systems is expensive in bandwidth usage, and it shows a serious lack in scalability. Also, more efficient systems based on distributed hash tables (DHT) lack in query expressiveness and flexibility. This paper addresses this issue by discussing existing solutions, and proposing a novel approach to support advanced multi-keyword queries in the context of Peer-to-Peer systems. It extends the existing, and widely established, DHT-based localization frameworks. This new approach can substantially reduce the bandwidth consumption and improve the load balancing over the network.

Keywords: Peer-to-Peer systems, Distributed Hash Tables, Content discovery, Keyword-based Searching.

1 Introduction

These last years have seen the emergence and the deployment of a new approach for distributed systems known as peer-to-peer. This approach allows the deployment of distributed systems in large-scale and dynamic networks. Nowadays, the main application of this approach is the object localization in these large scale networks. Objects can be of any type, such as files, services, applications, devices, etc. Though they have introduced a new manner to handle the resource discovery problem, the initial systems had some major limitations. One can mention Napster and its central index that introduces a bottleneck in the network and, therefore, a loss of reliability in addition to high administration costs. In the case of the Gnutella system [6], its expensive diffusion principle introduces a serious load in the network, as well as a serious lack in term of scalability.

To improve the efficiency of the classical approaches, distributed Hash Tables (DHT) have been introduced [16] [12] [17] [10]. The objective of the DHT-based peer-to-peer systems is mainly the localization of files (i.e. the node containing the requested file). These localization systems are based on the construction of a simple index containing associations between File IDs and Node IDs, where peers and data

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 630 – 641, 2006.

are structurally organized. This index is distributed over the network by the mean of a specific hash function. The DHT-based systems guarantee an efficient discovery with a limited number of hops. Furthermore, the study presented in [9] has demonstrated that DHT-based protocols are suitable for dynamic and large environments.

Despite the efficiency of these approaches, objects can only be localized using a unique identifier. Thus, current DHTs are limited to pure lookup of these identifiers. This unique ID introduces a problem as a user is not always aware of its value (i.e. the name of the corresponding file). However, in order to use these systems in the general case of resource discovery, it is necessary to define an enhanced search approach not only based on a unique ID but on several parameters, possibly fuzzy parameters, that can describe this resource (keyword searching). A system based on this principle will provide a query engine allowing this type of requests in large scale networks.

The objective of this work is to propose and evaluate a new solution to advanced multi-keyword search in the context of structured peer-to-peer systems. The related architecture is completely distributed and is based on DHT-localization. However, it is not based on a specific protocol but aims to incorporate, with minimal efforts, several protocols such as Tapestry [17] and Chord [16]. To support keyword-based requests, the proposed solution introduces a query engine on the top of these DHT-based protocols.

The rest of this paper is organized as follows. Section 2 provides and discusses related works. In section 3, we introduce our novel architecture. The sections 4 and 5 detail our propositions as well as the underlying mechanisms. We evaluate the solution in section 6 before we conclude with section 7.

2 Related Work and Discussion

Some work has already been done to make peer-to-peer keyword searching feasible. Most of the proposed solutions use the concept of reverse hash tables. The association < ID of file, Node > is replaced by the inverted list: < keyword, List of nodes >. Each resource is described by a list of keywords. Then, each keyword is indexed separately. Hence, the inverted index is distributed among peers by keyword. A query with *k* keywords can be answered by *k* nodes. Afterwards, all the results are collected by the initiator of the query and the final result is identified as the intersection of all these responses. Despite its simplicity, we can easily notice the overload introduced in the network by this approach, since the final result corresponds only to a small portion of the received responses. Based on this method, several recent propositions and improvements have been proposed. We can cite:

Reynolds and Vehdat [13] have proposed an architecture based on reverse hash tables associated with Bloom filters and caches to reduce network traffic.

Balazinska et Al. [1] have designed a resource discovery system, called *Twine*, based on the Chord [16] localization protocol. In this system, the support of keyword-based queries is achieved by the translation of resource descriptors into hierarchical trees (dependence between resources' attributes). This relation aims to reduce the load during the creation of the reverse hash tables.

Shi et Al. [15] used, also, the concept of reverse hash tables (keyword indexing). However, this mechanism is improved by organizing the nodes into several groups of different levels depending on their locations. This method aims to reduce the query routing latency as well as the network load.

Even so, the "*reverse hash table*" approach introduces a significant load in the network and nodes. In fact, each resource - and then each query - can be represented by potentially a number of keywords. Moreover, this approach raises the problem of "common keyword" which produces a heterogeneous load in the network. Thus, the node responsible for this so called "common keyword" will be requested more frequently than others, and consequently will be overloaded. The scalability limitations of this technique and its existing optimizations, in term of high bandwidth consumption, have been demonstrated in [14].

Other interesting works presented in [7] [11], introduce new concepts and architectures that are not compatible with existing P2P systems which is not the approach we have chosen. In fact, our objective is to exploit and extend the existing peer-to-peer systems, since they are widely used and accepted. Also, the proposed solution should provide a generic framework, which can be used for various purposes and applications such as: service discovery, file sharing, distributed file storage …etc.

Therefore, through this work, we aim to propose a new technique to handle advanced multi-keyword lookup queries in a large scale peer-to-peer environment. Indeed, this new approach intends to tackle the following objectives and requirements:

– **Generic framework:** use and extend the existing, and widely used, peer-to-peer frameworks. Thus, the new approach should be compatible with the current technologies.
– **Bandwidth saving:** reduce the bandwidth consumption to its minimum in order to improve scalability.
– **Load balancing:** reduce the load disparity between peers. However, this latter is not our main goal, and even if our proposed approach can reduce the load unfairness in the network, it still needs the introduction of more specific load balancing techniques [5] [2] in order to be completely efficient.

Hence, after describing our novel approach, we will evaluate its performances and compare it to the existing ones in order to prove these enhancements.

3 Proposed Architecture

In our solution, the system is deployed in a distributed manner on top of a set of participating nodes that communicate together using a peer-to-peer protocol. Each node supports the proposed software architecture as presented in Figure 1. The enhanced query layer is responsible for handling application requests and translating them into localization queries. This layer is deployed on the top of a DHT-based protocol such as Tapestry or Chord to take benefit from their routing and resource localization mechanisms. In the following part of this paper, we will present the various query mechanisms, as well as their integration with the content-localization protocol.

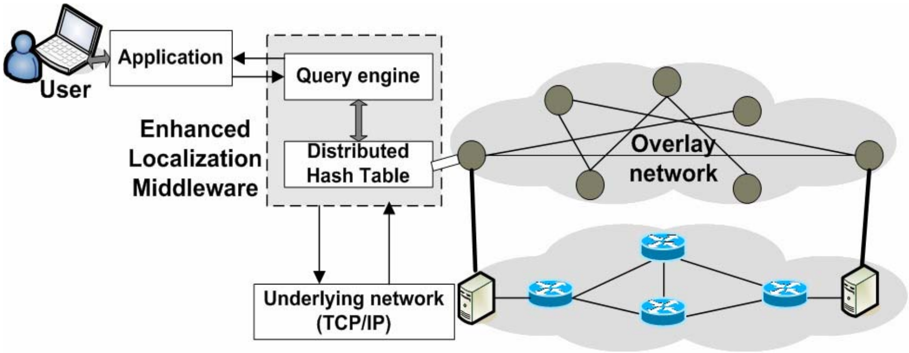


Fig. 1. Software architecture on a node

4 The Query Engine

The query engine constitutes the upper part of our discovery system. It receives high level search queries from the client application. Then, it translates them into routable queries which are forwarded to the underlying DHT-based localization layer. Afterwards, it analyses and combines the collected responses before delivering an accurate result to the application. So, the primary purpose of this engine is to provide advanced resource descriptions and a powerful query construction allowing the use and combination of various keywords.

4.1 Identifier Format and Resource Descriptions

In the current content-localization systems, the resource identifiers are obtained by hashing an attribute or a list of attributes using a consistent function. This list of attributes defines a single key that identifies uniquely the resource. Thus, if a resource is described by a key:

key = (attributeN°1=value1, attributeN°2=value2),

Then, its identifier will be extracted as follows:

ID = *h(key)*, where *h* represents the hash-function.

The naming space should be very large (generally 160 bits) in order to guarantee the identifier uniqueness with a high probability (consistent-hashing characteristic). In this case, the most used hash-function is SHA-1 [3].

The weakness of such localization systems is their incapacity to deal with complex and advanced queries. This comes from the specification of the Ids. In fact, each resource is identified by its unique key, instead of a list of attributes, which limits considerably its description. In order to respond to this lack, we propose to change the identifiers structure. This latter will be decomposed into several fields. For each resource type, a list of major attributes is established (attributes that should appear in the resource ID). This list should be larger than the key-attributes list and should model the resource in an accurate manner. Thus, the identifier is not based any more

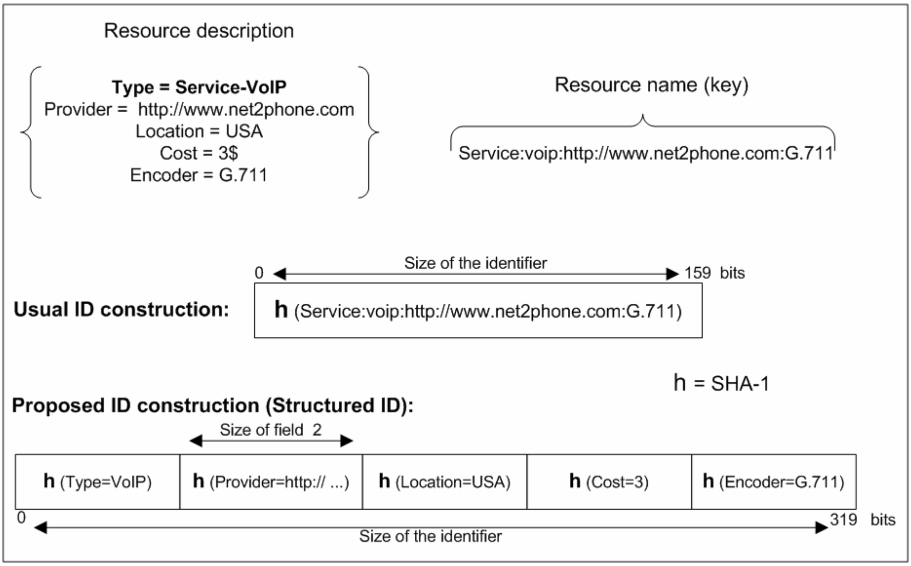


Fig. 2. Identifier format, usual vs. proposed method

on the resource key, but on its description (which is as complete as possible). The final identifier is specified according to the format presented in figure 2. This latter provides a comparison between the usual identifier construction technique and the proposed method.

The hash-function, that is used for each couple (attribute = value), is SHA-1. The first attribute should be the 'resource type', because it is the one that infers the exact identifier structure. In fact, the number of fields composing an identifier may vary according to the resource type (the list of major attributes describing a resource). Also, the sizes of the different fields can be different in a same ID. The unique constraint is to choose each field size proportionally to the Base B of the underlying localization protocol. This constraint aims to facilitate the routing process. Therefore, it is easy to combine the number and the size of the fields in order to comply with this constraint. For example, if a resource is described by 7 attributes and if the total identifier size is 320 bits, one solution is to fix the size of the first field to 80 bits and fix all the remainder to 40 bits. Naturally, the total ID size is the same for all the resources. This size should be very large in order to guarantee uniqueness with a high probability.

Concerning the node identifiers, we keep the same construction technique as in the existing systems, i.e. hashing of the IP address, the public key or any other unique attribute of the node.

4.2 Query Construction

Traditionally, in the content-localization systems, the requests are built by specifying the unique key of the desired resource (its identifier). Then, this request is routed to the node indexing this ID. The localization process is completed.

In our discovery system, the user application provides a set of keywords to the underlying query engine in order to formulate its request. Generally, these keywords are only a subset of all the attributes identifying the resource. Hence, the query engine can not, from this subset of attributes, recover a unique identifier to locate a resource. However, it can calculate some of its fields to build a fuzzy identifier (a set or interval of identifiers). Then, the localization protocol diffuses the request to all the nodes indexing the elements of this set of IDs. It is the concept of "limited diffusion". Figure 3 provides an example of such a query construction, based on our "*Fuzzy-identifier*" concept.

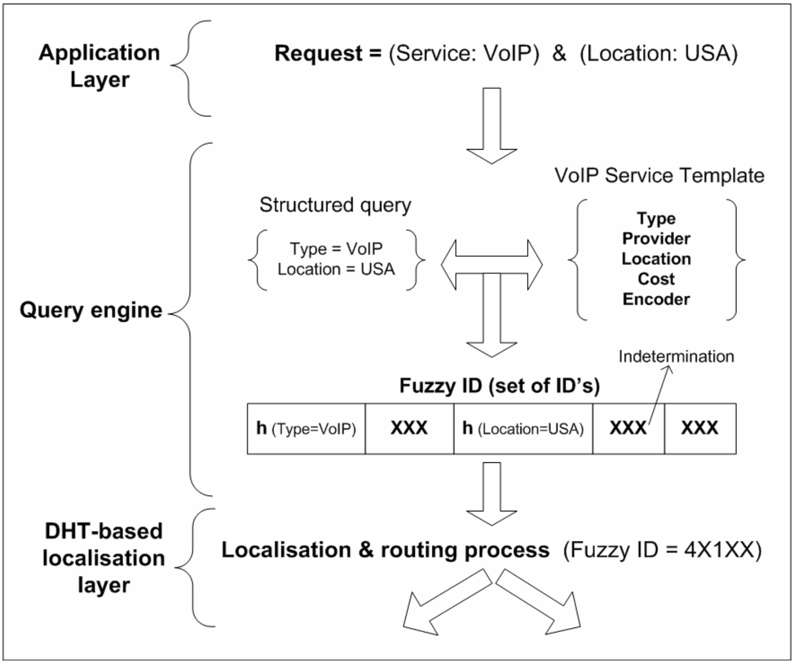


Fig. 3. Query construction example: the "Fuzzy-identifier" concept

When the user request combines several logical operators: *OR* / *AND*, the query engine translates them according to a canonical template:

$$(\mathbf{a} \,\&\, \mathbf{b} \,\&\, \mathbf{c}) \;||\; (\mathbf{d} \,\&\, \mathbf{e} \,\&\, \mathbf{f} \,\&\, \mathbf{g}) \;||\; \ldots\; (\mathbf{z} \,\&\, \mathbf{f}).$$

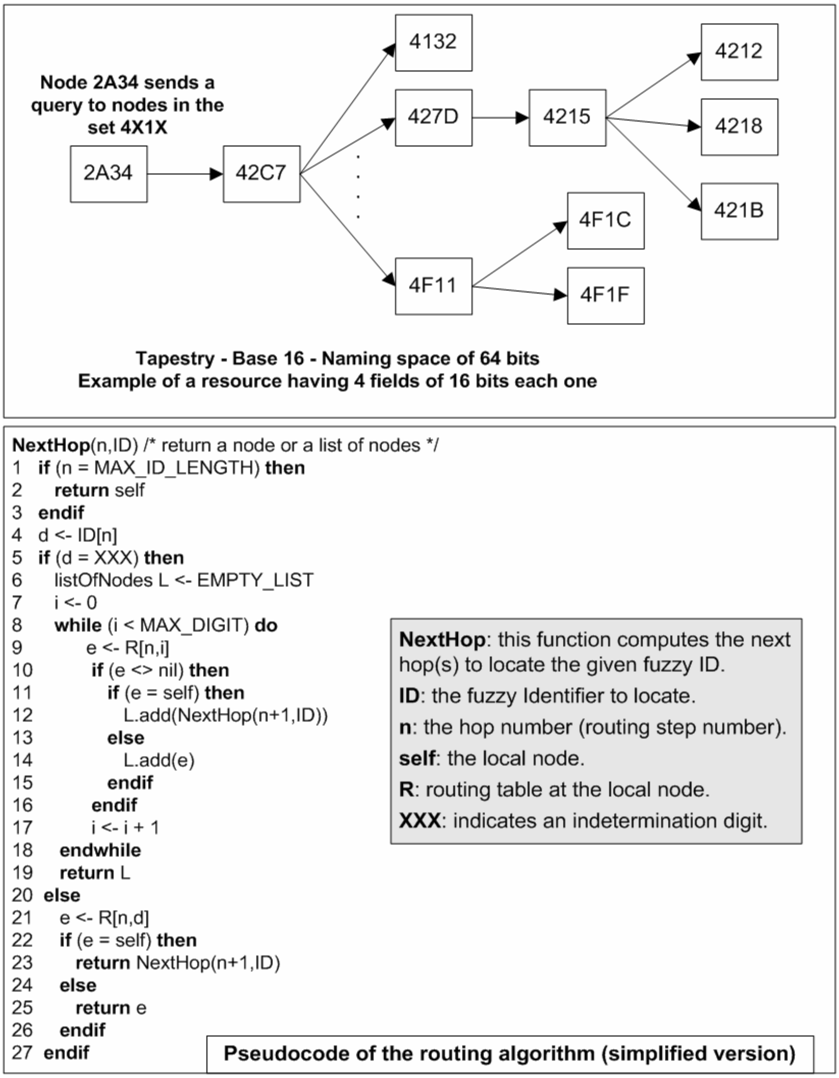
Where { **a**, **b**, …, **z** } represents query axioms. For example:

a = "Type=VoIP", **b** = "Location=USA", **c** = "Encoder=G.711", ...

Then, the query is divided into several basic queries according to the union operator. A basic query should contain only intersection operators. Then, each basic query is translated into a fuzzy identifier (an interval or a set of IDs). Thus, each fuzzy identifier constitutes a query which is sent to the localization layer in order to be routed. After receiving the responses, the query engine collects these results, combines and analyses them in order to extract the final result.

5 Query Routing: “Limited Diffusion”

When a resource is published and indexed in the network, its identifier is calculated using its major attributes. Then, the routing layer forwards this ID using the habitual process. However, during the search phase, the query engine provides to this localization layer a set of IDs. At that time, the "limited diffusion" mechanism is solicited, since the requesting node should contact several nodes simultaneously. In figure 4, we describe, as an example, the extension of the Tapestry protocol with this diffusion mechanism.


Node 2A34 sends a query to nodes in the set 4X1X
2A34
42C7
4132
427D
4215
4212
4218
421B
4F1C
4F11
4F1F
Tapestry - Base 16 - Naming space of 64 bits
Example of a resource having 4 fields of 16 bits each one

```
NextHop(n,ID) /* return a node or a list of nodes */
if (n = MAX_ID_LENGTH) then
  return self
endif
d <- ID[n]
if (d = XXX) then
  listOfNodes L <- EMPTY_LIST
  i <- 0
  while (i < MAX_DIGIT) do
      e <- R[n,i]
      if (e <> nil) then
        if (e = self) then
          L.add(NextHop(n+1,ID))
        else
          L.add(e)
        endif
      endif
      i <- i + 1
  endwhile
  return L
else
  e <- R[n,d]
  if (e = self) then
    return NextHop(n+1,ID)
  else
    return e
  endif
endif
```

NextHop: this function computes the next hop(s) to locate the given fuzzy ID.

ID: the fuzzy Identifier to locate.

n: the hop number (routing step number).

self: the local node.

R: routing table at the local node.

XXX: indicates an indetermination digit.

Pseudocode of the routing algorithm (simplified version)

Fig. 4. Limited diffusion with Tapestry

In the Tapestry protocol [17], every node and every resource is assigned a unique ID. Indeed, the identification space is structured in a hierarchical tree. Thus, the routing is performed in a recursive manner by progressing digit by digit in the targeted node ID, using the routing table entries of each visited node. In the case of our extension, the routing algorithm handles the "*undetermined digits*" by forwarding the request to all the relevant entries in the routing table, as shown in figure 4. Hence, it is a *hybrid* solution (routing vs. diffusion). This extension introduces only minor changes into the functioning (routing algorithm [17]) of this localization protocol. All the other features and algorithms of the Tapestry protocol are kept unchanged, especially in term of replication, fault tolerance, network construction and maintenance. In the same way, our architecture can use and extend other DHT-based localization protocols, such as Chord [16] or Pastry [19].

Also, the localization protocol keeps his performances unchanged. In order to confirm this assertion, we initiated a set of tests, using the p2psim software [4], and concerning our extension of the tapestry protocol. The simulation consists on varying the size '*N*' of the network and the base '*B*' of the protocol (the base of the identification space). The curves in figure 5 exhibit clearly the O(Log$_B$ N) characteristic of the query routing algorithm, in term of mean hop count (the number of hops necessary for a query to reach all its destinations).

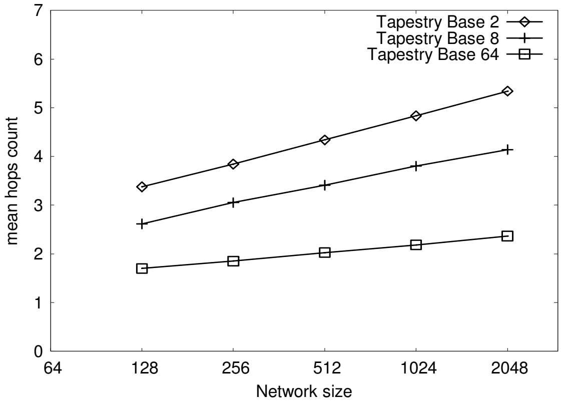


Fig. 5. Scalability Performance of the routing algorithm

6 Evaluation of the Query Engine

In this section, we evaluate the performances of the proposed query engine by comparing our approach, the *fuzzy identifiers* technique, to the *reverse hash table* approach. This latter is described in section 2: Related work. This comparison is mainly based on simulations and made with respect to three crucial features:

- **Storage cost:** by computing the mean index size per node.
- **Load balancing:** by tackling the load disparity between nodes in term of index distribution.

- **Bandwidth cost:** by computing the mean number of messages received in response to a search query.

Before going further in this evaluation, we define in Table 1 a set of experiment parameters and metrics, as a basic *vocabulary* for the rest of this section. Thus, the system has *N* peers. The overall index is composed of *M* resource descriptions and distributed across the network peers. Each resource is described using *K* relevant keywords.

Table 1. Parameters and metrics

Name	Description
N	Number of nodes in the network.
M	Number of documents (resource descriptions) in the network.
K	Mean number of keywords to describe a resource, or document.
r	Replication factor (index replication for availability purposes).
$\mathbf{D_i}$	Size of the index maintained by node$_i$.
D	Mean index size per node.
L	Load disparity factor (in term of index distribution)

The mean index size per node determines the storage cost to distribute and maintain the index over the network peers. This metric is obvious and easy to compute in both cases:

Fuzzy identifiers:
$$D = \frac{r \times M}{N} \tag{1}$$

Reverse hash table:
$$D = \frac{r \times K \times M}{N} \tag{2}$$

Indeed, in our approach, each resource is described using only one index entry. On the other hand, in the reverse hash table approach, each keyword of the resource is indexed separately (the K factor in the second formula). We can notice the big storage loss due to the use of reverse hash tables in comparison to our approach, since the mean number of keywords per resource is generally big enough.

Another critical issue of the index distribution performance is the load balancing (index balancing over the network peers). In order to study this feature, we defined a specific and accurate metric. This metric is computed as the *standard deviation* of the discrete quantitative variable: *($S_i = D_i / D$)*. By definition, the mean value of S_i is 1. Also, the mean index size has no effect on this variable, and thus on its standard deviation. Therefore, this metric reflects only the index load disparity between nodes. In an ideally balanced system, the metric should be equal to zero. Also, high values imply an unfair index distribution as well as the presence of extremely loaded nodes. We call this metric *"Load disparity factor"*:

$$L = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} \left(\frac{D_i}{D} - 1\right)^2} \qquad (3)$$

Figure 6 compares our approach to the reverse hash table technique. In these experiments, the overall number of documents, *M*, is set to one million, and the mean number of keywords, *K*, is set to 8. The index is not replicated ($r = 1$). First, we define the global keyword and document space, reflecting a realistic situation. Then, inside this space, we chose randomly the set of *M* documents to constitute our index. This latter is distributed across the network hosts according to both techniques. The curve in figure 6 shows the effect of the network size on the load balancing metric. The use of a reverse hash table leads to a highly unbalanced index distribution. Furthermore, the network size has a big effect on the load disparity parameter, which is very constraining for a potentially large scale localization system. On the other hand, our approach showed low and quite stable results. Thus, it is more suitable for a deployment in large scale environments. However, these results are not optimal. The system needs the introduction of more specific load balancing techniques [5] [2] in order to improve its performances.

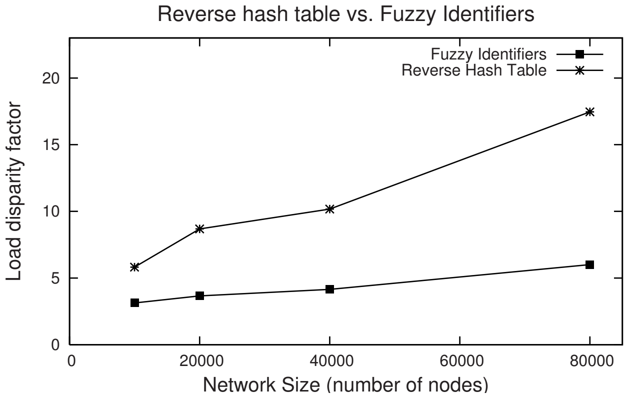


Fig. 6. Load disparity between nodes: the network size effect

The number of documents received as a response to a search query is a critical issue of a scalable P2P localization system, since it has a big impact on both, the aggregate bandwidth cost and the processing load on peers. This critical parameter is plotted in figure 7 as a function of the query accuracy. By this latter, we mean the number of specified keywords in the query. If a query is more precise, by providing several keywords, it should produce fewer responses. Hence, the mean number of received documents should be inversely proportional to the query accuracy. This assertion is verified in the case of our approach. Furthermore, this approach is optimal since the number of received responses corresponds to the final result of a query; the bandwidth consumption is reduced to its minimum.

In the case of the reverse hash table approach, each keyword is indexed separately; so, a query with *k* keywords is divided into k requests and answered by *k* nodes. Then, all the results are collected and the final result is the intersection of all the received responses. Therefore, the mean number of received documents is proportional to the query accuracy, while the final result is inversely proportional to this parameter, which is awkward. Moreover, as shown by figure 7, the use of a reverse hash table leads to a huge waste of bandwidth. The use of Bloom filters and cache methods [13] may reduce this loss; but it is only a partial enhancement and not a complete solution to this problem.

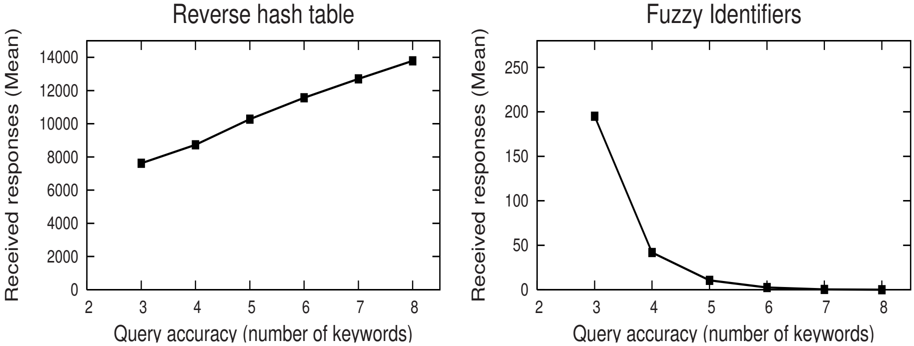


Fig. 7. Bandwidth cost per query: the query accuracy effect

This waste of bandwidth appears again during the registration phase. In fact, when a new resource is indexed in the system, a registration message is sent for each of its keywords. While, in our approach, only one message is produced and sent during the registration phase, which is more efficient.

10 Conclusion

Advanced keyword searching is not feasible in large scale networks using the usual *reverse hash table* approach, because of its various limitations. For that reason, we have proposed and detailed a new solution to handle this important requirement. This solution exhibits the following strengths: it extends the efficient DHT-based peer-to-peer frameworks, in a generic way; it minimizes the bandwidth and storage costs; and it reduces the unfairness related to the index distribution. All these assertions have been demonstrated through various simulations and experiments. Despite these improvements, the index distribution problem is still open and therefore we are investigating an efficient load balancing algorithm in order to improve.

As a future work, we will use our enhanced localization system in a global service discovery architecture for large ambient networks, where the number of resources can be extremely large and dynamic. This architecture should overlap the existing local service discovery systems.

References

1. Balazinska M., Balakrishnan H., Karger D., « INS/Twine : A scalable peer-to-peer architecture for intentional resource discovery», *Pervasive 2002 - 1st International conference on Pervasive computing,* Zurich, Switzerland, 26-28 August 2002.
2. Byers J., Considine J., Mitzenmacher., « Simple load balancing for distributed hash tables », *Proceedings of the 2nd IPTPS*, Berkeley, CA, USA, 20-21 February 2003.
3. FIPS 180-1., « Secure hash standard » U.S. Department of Commerce/NIST, Springfield, VA, April 1995.
4. p2psim, http://pdos.lcs.mit.edu/p2psim/.
5. Godfrey B., Lakshminarayanan K., Surana S., Karp R., Stoica I., « Load balancing in dynamic structured P2P systems », *Proceedings of IEEE INFOCOM 2004*, Hong Kong, 7-11 March 2004.
6. Gnutella, http://gnutella.wego.com/ .
7. Heng Tao Shen, Yanfeng Shu, Bei Yu, «Efficient Semantic-Based Content Search in P2P Network» *IEEE Transaction on Knowledge and Data Engineering*, No. 7, July 2004, pp. 813-826.
8. Harren M., Hallerstein M., Huebsch R., Thau B., Shenker S., Stoica I., « Complex queries in DHT-based peer-to-peer networks », *Proceedings of the 1st IPTPS*, Cambridge, MA, USA, 7-8 March 2002.
9. Li J., Stribling J., Kaashoek F., Morris R., Gil T., « A performance vs. cost framework for evaluating DHT design tradeoffs under churn », *Proceedings of IEEE INFOCOM 2005*, Miami, FL, USA, 13-17 March 2005.
10. Meymounkov P., Mazieres D., « Kademilia : A peer-to-peer information system based the XOR metric », *Proceedings of the 1st IPTPS*, Cambridge, MA, USA, 7-8 March 2002.
11. Mischke J., Stiller B., « Rich and Scalable Peer-to-Peer Search with SHARK », Autonomic Computing Workshop, Fifth Annual International Workshop on Active Middleware Services (AMS'03), 25 June 2003.
12. Ratnasamy S., Francis P., Handley M., Karp R., Shenker S., « A scalable content-addressable network », *Proceedings of SIGCOMM 2001*, San Diego, CA, USA, August 2001, p. 161-172.
13. Reynolds P., Vahdat A., « Efficient peer-to-peer keyword searching », *Proceedings of ACM/IFIP/USENIX Middleware 2003*, Rio De Janeiro, Brazil, 16-20 June 2003.
14. Jinyang Li, Boon Thau Loo, « On the Feasibility of Peer-to-Peer Web Indexing and Search », *Proceedings of the 2nd IPTPS*, Berkeley, CA, USA, 20-21 February 2003.
15. Shi S., Yang G., Wang D., Yu J., Qu S., « Making peer-to-peer searching feasible using multi-level partitioning », *Proceedings of the 3rd IPTPS*, San Diego, CA, USA, 26-27 February 2004.
16. Stoica I., Morris R., Karger D., Kaashoek F., Balakrishnan H., « Chord : A scalable peer to peer lookup service for internet applications », *Proceedings of SIGCOMM 2001*, San Diego, CA, USA, August 2001, p. 149-160.
17. Zhao B., Huang L., Stribling J., Rhea S., Joseph D., « Tapestry: A Resilient Global-Scale Overlay for Service Deployment », *IEEE Journal on Selected Areas in Communications*, vol. 22, n° 1, January 2004, p. 41-53.
18. Rodrigues R., Liskov B., « High availability in DHTs: Erasure coding vs. Replication », *Proceedings of the 4th IPTPS*, Ithaca, NY, USA, 24-25 February 2005.
19. Rowstron A., Druschel P., « Pastry: Scalable, distributed object location and routing for largescale peer-to-peer systems », *Proceedings of ACM/IFIP Middleware 2001*, Heidelberg, Germany, 12-16 November 2001.

Chasing: An Efficient Streaming Mechanism for Scalable and Resilient Video-on-Demand Service over Peer-to-Peer Networks*

Jian-Guang Luo, Yun Tang, and Shi-Qiang Yang

Tsinghua University, Beijing 100084, P.R. China
{luojg03, tangyun98}@mails.tsinghua.edu.cn,
yangshq@mail.tsinghua.edu.cn

Abstract. Provisioning scalable and resilient Video-on-Demand (VoD) service is both challenging and interesting. Recently, peer-to-peer (P2P) networks are introduced to address the scalability of VoD service over Internet. Most of existing work follows the line of cache-and-relay (CR) scheme to accommodate the asynchronous characteristic of requests from a community of end users. Aiming to take full advantages of bandwidth capacities at each node and pre-recorded feature of requested video files at streaming server, we improve traditional CR approach by efficiently exploiting surplus bandwidth and proactively prefetching media contents from either the server or other peers. Our proposed basic chasing and advanced chasing mechanism not only achieve significant reduction of workload on streaming server, which could translate into better scalability, but also help the streaming session to adapt to volatile network fluctuation. Our extensive experiments have demonstrated encouraging results with respect to increased system performance.

1 Introduction

With the growth of Internet, many researchers have recognized the potential of providing video-on-demand (VoD) service to a large population of network users. In traditional client/server approach, each client subscribes streaming service directly from the server through unicast connections. Due to high bandwidth consumption and long-playing features of streaming sessions, the server soon becomes the bottleneck of the system as the popularity increases. Therefore, a number of IP multicast based solutions, comprising batching [8], patching [9], periodical broadcasting [10], stream merging [11], and etc., have been proposed. However, they are infeasible to deploy in the absence of network infrastructure support. Other efforts advocate proxy-based approaches [12] or CDN [13] to balance server workload by deploying a number of secondary servers at the edge of network. However, it is difficult to place the servers efficiently over Internet due to the high dynamic distribution of clients. Furthermore, the exorbitant cost also affects the wide deployment of proxy-based solutions or CDN.

* This work is supported by the National Natural Science Foundation of China under Grant No.60432030 and No.60503063.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 642–653, 2006.

Fortunately, peer-to-peer (P2P) architecture is introduced into the arena of large scale data dissemination in recent years. In a typical P2P environment, each node plays the role of both server and client, contributing its available computation, storage and/or bandwidth resources into the collective resource pool. As the benefits accrue with mutual cooperation among peers, P2P community offers a scalable approach to support a large number of users without either costly servers or network infrastructure support. Therefore, P2P networks are promising to solve the scalability problem in large scale VoD applications.

Despite the success of P2P live video streaming [5, 6, 7], P2P VoD services are of significant differences. Most importantly, intrinsic to the notion of on demand, the video viewed by different users may vary in file position. This asynchronous nature of VoD might be regarded as running counter to the fundamental design philosophy of P2P networks. Besides, compared to live video streaming, the requested video are always pre-recorded in typical VoD systems, and this ”***forward-availability***” enables appealing features of content prefetching. Most of existing work follows the line of cache-and-relay (CR) scheme [1, 2] to tackle the asynchronous problem, in which end nodes do not prefetch any contents in advance, and thus the visual quality at users might suffer severely from network fluctuations. Another latest proposed protocol dPAM [3] utilizes the surplus downloading bandwidth of each peer to prefetch some portions of video content, so it really alleviates the disruption of video playback under dynamic network fluctuation and provides each node capabilities to recover from upstream nodes failure or departure. Nevertheless, as soon as the incoming bandwidth of peers is lower than twice the video playback rate, which is fairly common in realistic networks, dPAM will cause much more users to directly stream from source server, which in turn imposes higher workload than CR scheme.

In this paper, we try to take full advantages of bandwidth capacities at each node and pre-recorded feature of requested files at streaming server to achieve better system scalability and resilient visual QoS. Our main contributions lie in twofold. First, by efficiently exploring surplus bandwidth, we propose basic chasing to greatly reduce the server bandwidth cost. Second, we enforce proactive prefetching from either the server or other peers, namely advanced chasing to ensure that all peers are resilient to volatile network fluctuation.

The remainder of the paper is organized as follows: Section 2 gives a brief introduction to CR and dPAM and further derive our motivations. Section 3 presents the proposed chasing mechanism in detail and we intentionally separate it into basic and advanced chasing with respect to different motifs. Simulation results of experiments and comparisons are provided in Section 4. At last, we conclude the paper and discuss future work in Section 5.

2 Cache-and-Relay and Prefetching

In this section, the key principles and drawbacks of CR and prefetching schemes are briefly analyzed to further concentrate our motivations. In the context of this paper, we only consider the representative case of CBR media distribution with single source server. For sake of exposition, the playback rate of video stream is assumed to be r_p, while the downloading rate of client is denoted as r_d. The maximum downloading bandwidth capacity of each peer is assumed to be r_c identically. Thus, we have the following relations: $r_c \geq r_p$, $r_d \leq r_c$. In the rest of the paper, we use R_1, R_2 and R_3 to denote the sequential requests or clients which arrive at time t_1, t_2 and t_3. And the buffer size of R_1, R_2 and R_3 is w_1, w_2 and w_3 time units respectively. Each peer in P2P networks is able to buffer the streamed content for a certain amount of time and overwrite its buffer window in a circular manner.

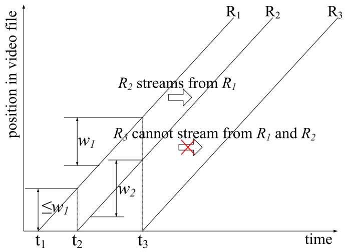


Fig. 1. Traditional cache-and-relay scheme: illustrative example

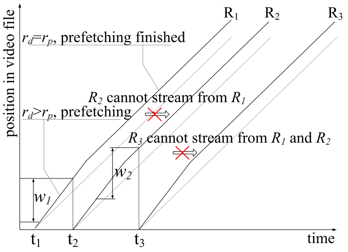


Fig. 2. Prefetching scheme in dPAM: illustrative example($\alpha = 1.33$)

The CR scheme is illustrated schematically in Fig.1. As shown, when R_2 demands the service at t_2, the contents desired by R_2 are available within the buffer of R_1 because the gap between R_1 and R_2 falls into the size of R_1's buffer, that is, $t_2 - t_1 \leq w_1$. Hence, R_2 starts streaming directly from R_1 instead of going to the server. However, when R_3 joins the service community at t_3, it could not find appropriated upstream clients since the requisite contents has already been drained out from the buffer of both R_1 and R_2. Here R_3 will seek to the support of the source server. Although CR has been demonstrated to scale well in [1], the downloading rate of peers is always equal to the playback rate, which means all the peers attempt to play the newest content in their buffers. Therefore, the visual quality of playback is highly sensitive to network fluctuations and the departure or failure of upstream peers.

Prefetching scheme is proposed in dPAM [3] to alleviate QoS degradation with the consideration of "forward-availability". Fig.2 exhibits the examples of prefetching, in which each client tries to fill its own buffer at the maximum downloading rate. The solid line depicts the position of the downloading content, while the dotted line shows the position of content being playback at the end node. The fundamental advantage of prefetching is to store more contents by exploiting surplus bandwidth of clients and hence offering an increased service robustness to network dynamics. However, it could also be observed that R_2 is not able to stream from R_1 anymore under the same arrival pattern in Fig.1. This is because the content cached in the buffer of R_1 is refreshed at a faster rate than in CR scheme. In such a case, R_2 has to establish a new connection to the server if the maximum downloading rate r_c is smaller than twice the playback rate R_p. To make the question clear, we define $\alpha = r_c/r_p$, which could be utilized as the indication of prefetching capabilities. We could deduce that dPAM will impose more bandwidth consumption upon server even than CR scheme when $\alpha < 2$. In Fig.2, $\alpha = 1.33$.

Essentially motivated by above analysis, in next section we mainly consider how to effectively utilize surplus bandwidth capacity of peers to improve system properties in terms of reduced server workload and higher service resilience. Our objective comes in twofold. One is to further reduce bandwidth cost of server to keep the system scalable. The other is to eliminate QoS degradation of video playback due to network fluctuations and parent switch operations by prefetching portions of contents.

3 Chasing Mechanism

In this section, we depict chasing mechanism in two steps. Firstly, in subsection 3.1, by efficiently exploiting surplus bandwidth capacity at each node, we improve traditional CR scheme by basic chasing mechanism, which significantly reduce server bandwidth cost. Secondly, in subsection 3.2, by proactively considering the "forward-availability" of requested files, we enable more end clients to prefetch contents by advanced chasing mechanism, and therefore alleviating visual quality degradation. It should be stressed that, in this paper we mainly concentrate our work in the context of $1 < \alpha < 2$. Otherwise each node could directly apply patching techniques by opening two streams simultaneously.

3.1 Basic Chasing Mechanism

The substantial drawback of traditional CR scheme lies in the striking workload imposed by "***lonely***" peers, who are either newcomers or foundling peers. Aiming to further reduce the consumption of server bandwidth, we intuitively resort to encourage more peers to take charge of the fostering procedure. In basic chasing mechanism, each peer attempts to catch up with the earlier ones which are already in the community. Compared to Fig.1, R_3 with basic chasing mechanism in Fig.3 is now able to stream from R_2 with a transient favor of server under the same arrival pattern. Hence the bandwidth consumption of server is diminished. The detailed discussions proceed along the dimensions of new arrival and parent failure or departure in following schematic analysis.

New Arrival: When the new coming peer R_3 enters the service community at time t_3, the oldest content in the buffer of R_2 is $w_d = (t_3 - t_2 - w_2) > 0$, leaving a gap of w_d time units to what R_3 demands. Thus R_3 establishes a streaming connection rated at r_p from server for these $[0, w_d)$ packets. At the same time, R_3 is still able to fetch portions of intending video contents in advance from R_2 with the aid of the surplus bandwidth $(\alpha - 1) \times r_p$, different from CR. This parallel downloading procedure essentially builds the most important component of basic chasing mechanism. Till $t_3 + w_d$, R_3 starts to playback the video stream from position w_d. Since R_3 has already cached parts of the stream from

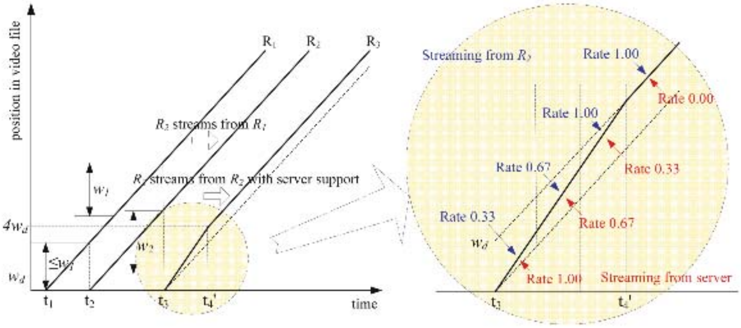


Fig. 3. Basic chasing: illustrative example($\alpha = 1.33$)

R_2 ranged in $[w_d, 2w_d)$, R_3 now only need to download the remaining stream rated at $r_p - (r_c - r_p) = 2r_p - r_c$ from server. Therefore, the streaming bandwidth from R_2 to R_3 currently increases to $r_c - (2r_p - r_c) = 2(r_c - r_p)$. If $2(r_c - r_p) \geq r_p$, i.e., $\alpha \geq 1.5$, R_3 is now able to download the entire stream from R_2. Consequently, at time $t_3 + 2w_d$, basic chasing mechanism terminates the parallel streaming process and from then on R_3 can be fully served by R_2. If $\alpha < 1.5$, the process will continue for a longer duration, and in each interval w_d, the bandwidth of R_3 spent to stream from R_2 increases by $(r_c - r_p)$. Since the bandwidth used by R_3 to prefetch content increases step by step, we call the chasing process "***step-like***". Basically, R_3 totally spends $ceil(1/(\alpha - 1)) \times w_d$ time units to catch up with R_2 altogether, resulting in a lower server workload after an initial acceleration.

Departure or Failure of Parent: When one node leaves the P2P community or experiences a transient failure, the streams to all of its children nodes halt. Then those downstream foundlings have to search new suppliers or alternatively ask the server for help. We keep in mind that if some peers have employed basic chasing at previous rounds, they should have some prefetched content in their buffers. So we depict an general non-overlapped buffer scenario of R_1 and R_2 in Fig.4, in which R_1 and R_2 has w_{p1} and w_{p2} time units of content prefetched in their buffers respectively, and the "missing" content between the buffer of R_1 and R_2 is w_d time units. Since the prefetched content and "missing" content (if downloaded) will be exhausted to playback in $(w_{p2} + w_d)$, the bandwidth used by R_2 to download the missing content from server should be at least $w_d \times r_p/(w_{p2} + w_d)$ to ensure the smooth playback. Then the chasing scheme can be understood in the following two cases:

1) If $w_d \times r_p/(w_{p2} + w_d) \leq (r_c - r_p)$, R_2 uses bandwidth of $w_d \times r_p/(w_{p2} + w_d)$ to fetch the "missing" content from server, and meanwhile the remainder bandwidth is still enough to prefetch the entire stream from the buffer of R_1.

2) Otherwise, since R_2 has to reserve the bandwidth of $w_d \times r_p/(w_{p2} + w_d)$ to ensure the smooth playback, R_2 only has bandwidth rated at $r_c - w_d \times r_p/(w_{p2} + w_d)$ to prefetch portions of stream from R_1, and is forced to start a "step-like" chasing process to catch up with R_1.

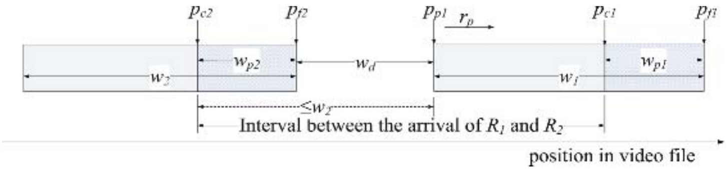


Fig. 4. Non-overlapped buffer scenario of R_1 and R_2 in basic chasing mechanism

Whether a peer just joins the service community or it experiences parents' failure or departure, the basic chasing mechanism described above could achieve lower server workload because the surplus bandwidth at each peer indeed helps server to share the responsibility to foster. We emphasize this salient feature of basic chasing as one of main contributions in this paper. Afterwards, several key components of proposed mechanism will be examined summarily.

Buffer Size: In basic chasing mechanism, the buffer size and the interval between the arrivals of peers definitely plays a key role in the decision to whether the former peers can be chased. We can image in Fig.4, if $w_2 < w_{p2} + w_d$, the buffer of chasing client R_2 could not cache the contents going to prefetch during the process of chasing R_1. Since we have assumed a sequential access model for client requests, given the buffer size w_1, w_2 of R_1 and R_2, the maximum interval between their arrivals is $(w_1 + w_2 - w_{p1})$ so that R_2 can stream from R_1. Recall that the maximum interval allowed in cache-and-relay scheme is w_1, only depending on the buffer size of R_1. We claim that basic chasing explores the utility of the buffer of the chasing clients, thus provides more clients with opportunities to share a single stream from server, and in turn effectively reduce the server bandwidth cost.

Downloading Bandwidth: We have assumed the maximum downloading bandwidth r_c of peer is ranged between the playback rate and twice of this rate, i.e. $1 < \alpha < 2$. Although it is clear that whether a client could chase an earlier node depends on the buffer states of clients, rather than the maximum downloading bandwidth, the downloading bandwidth capacity will definitely affect the chasing process due to the "step-like" characteristic. Clearly, the larger α is, the less steps chasing process needs and in turn less contents would be stream from server. Therefore, we claim that larger downloading bandwidth of peer will lead to lower server workload. This result will be clearly demonstrated in simulation experiments in Section 4.

Our proposed basic chasing mechanism is similar to the partition scheme of bandwidth skimming [4]. However, there are two distinguished differences between them. One is that proposed chasing applies to P2P environment while the partition scheme of bandwidth skimming is deployed in IP multicast. The other is that chasing streaming in general is deemed to catch up with the buffer of parents and it goals to achieve asynchronous multicast in overlay network. On the contrary, clients with partition scheme tries to follow the step of IP multicast stream, composing unique synchronous multicast eventually.

In this subsection, we mainly illustrate the basic chasing mechanism for newcomers and ones that suffer from parents' failure or departure. Additionally, some root parameters of this mechanism, e.g. buffer size and downloading capacity are also analyzed qualitatively. In next subsection, we further extend basic chasing mechanism to offer a more resilient streaming mechanism.

3.2 Advanced Chasing Mechanism: Proactive Prefetching

In basic chasing mechanism, the parallel downloading procedure aids a continuous playback for those who are often regarded as "***lonely***" peers. However, for other normal subscribers, if they could find appropriate upstream peers to obtain cooperative services when joining the system, all of them will receive the stream at the playback rate, so that they will suffer severe visual quality degradation from network fluctuation. Inspired by the prefetching in basic chasing mechanism, we import proactive prefetching into other peers and improve basic chasing scheme by enforcing them to cache some contents in advance from server or other peers. This so called advanced chasing mechanism ensures considerable resilience to network dynamics.

Fig.5 exhibits the advanced chasing mechanism explanatorily. Different from traditional CR scheme and basic chasing mechanism, advanced chasing tries to download the contents from either server or other peers at the maximum downloading bandwidth rate r_c once R_1 arrives in the system. As soon as the prefetched contents reach a predefined threshold, R_1 slows its downloading rate to the playback rate r_p and the primordial prefetching ceases. For the sake of explanation, here we define β as ratio of the contents that one is willing to prefetch to its own buffer size. As shown in Fig.5, R_1 decreases its downloading rate r_d to r_p once its prefetched content achieves $\beta \times w_1$ at time t'_1. Clearly, the quick buffer refreshment due to proactive prefetching in advanced chasing will impose an obstacle to the stream sharing between clients as in dPAM. At time t_2 the contents required by new participant R_2 have already been overwritten by the prefetched data in the buffer of R_1. Then R_2 cannot help but to attain what it needs from the server with dPAM scheme. However, R_2 in Fig.5 is still able to stream from R_1 with proposed basic chasing mechanism. Nevertheless, compared to basic chasing, if the forthgoer R_1 deploys advanced chasing mechanism to proactively prefetch, it is apparent that the chance for R_2 to chase R_1 is diminished.

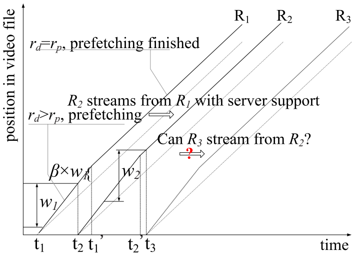


Fig. 5. Advanced chasing mechanism: illustrative example($\alpha = 1.33$)

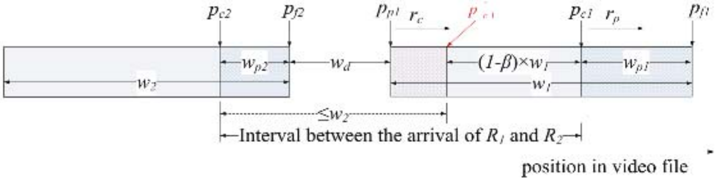


Fig. 6. Non-overlapped buffer scenario of R_1 and R_2 in advanced chasing mechanism

As mentioned above, with advanced chasing mechanism R_1 may be downloading contents at its maximum downloading rate r_c and refreshing its buffer equally when R_2 attempts to chase R_1, which is illustrated in Fig.6. Observe that R_1 and R_2 already have w_{p1} and w_{p2} time units of prefetched contents in their buffers respectively. Because the pointer p_{p1} is moving more quickly than the playback rate r_p, it is infeasible to adopt the basic chasing mechanism directly to obtain portions of the stream contents from p_{p1} at the buffer of R_1. Recall that R_1 will decrease its downloading rate to r_p afterwards as soon as its prefetched contents achieve $\beta \times w_1$, so henceforward R_2 is able to gain cooperative service from the buffer of R_1 at position p'_{c1} shown in Fig.6. Therefore, the interval between the arrivals of R_1 and R_2 should be smaller than $(1-\beta) \times w_1 + w_2$ to ensure that R_2 can be served by R_1. Evidently, the benefits among peers to share burden of server decrease slightly when compared to basic chasing, but we still argue that if the buffers are of same size, i.e. $w_1 = w_2$, the maximum arrival interval in advanced chasing is larger than that in cache-and-relay scheme if $\beta < 1$, which still provides increased performance in terms of reducing the server bandwidth in comparison with CR scheme.

Due to space limitation, we do not intend to present every detail within the advanced chasing mechanism. By similar reasoning, we could understand it with buffer size, downloading capacity and other complexities. Intuitively, in advanced chasing mechanism, all the clients will prefetch some contents due to the proactive prefetching process, so that they can conceal a temporary degradation or jitter of downloading bandwidth without affecting the playback quality. From the viewpoint of system performance, advanced chasing mechanism attemps to balance the reduction to server workload by basic chasing with the resilience level that prefetching provides. In the face of extreme scenario, basic chasing could be thought as the case with $\beta = 0$ of advanced chasing mechanism.

4 Performance Evaluation

In this section, we evaluate the performance of proposed chasing mechanism by extensive simulations. We first explain the setup of our simulation in subsection 4.1, and then quantify the comparison between basic chasing and CR in subsection 4.2, and in subsection 4.3 we compare the advanced chasing with both dPAM and basic chasing with respect to the amount of prefetched contents.

4.1 Simulation Setup

We implemented a discrete time driven tool to simulate basic chasing, advanced chasing, as well as cache-and-relay and dPAM mechanisms. To make the question clear, we reasonably simplify the heterogeneity of end hosts by assuming they are with identical buffer size w, maximum bandwidth capacity ratio α and desired ratio of prefetched contents β. Since most of existing P2P applications always provision substitutes foreseeingly, it is also assumed that each peer could determine where it should obtain service without any delay when it suffers the departure or failure of parents.

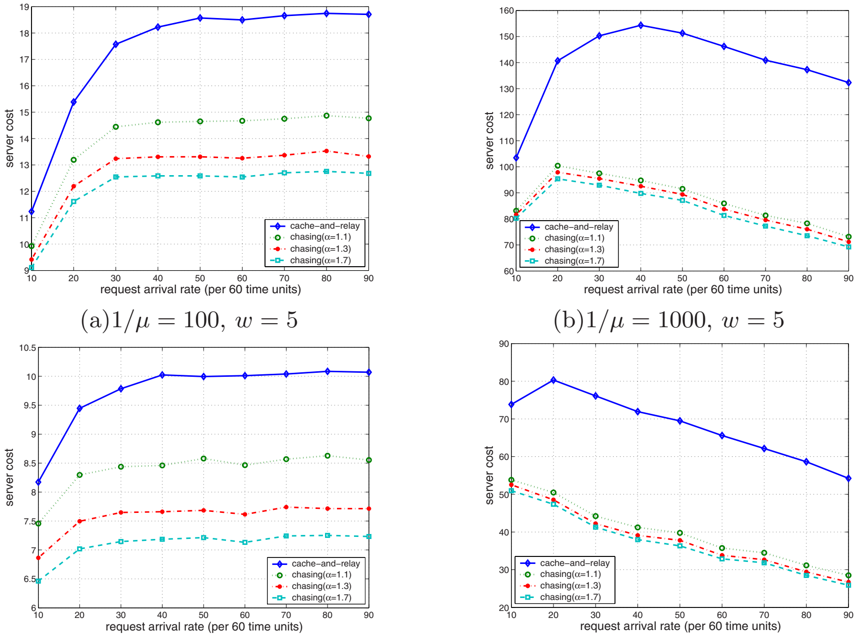
In simulation, request arrival pattern is produced according to a standard Poisson process with rate λ and the duration of clients staying in the community follows an exponential distribution with rate μ. It should be mentioned that the online user number will increase continuously at the very beginning of the simulation and then come into a steady state after a while. Here we utilize the average ratio of server's total outflow traffic to one single flow at playback rate to evaluate the server bandwidth consumption. This metric indicates how many streams flow out from the server and, more weightily, also take the duration of stream session into consideration. Each point on the following figures represents an average over 10 independent runs.

4.2 Performance of Basic Chasing Mechanism

We design experiments to investigate the performance comparison between basic chasing mechanism and cache-and-relay in terms of server bandwidth consumption and the simulation result are illustrated in Fig.7(a)-(d) with different average online duration $1/\mu$ and buffer size w, respectively. Observe that whatever these parameters are, the server bandwidth consumption in base chasing mechanism is always lower than CR scheme. Also it declines with the increase of the downloading bandwidth capacity α. Towards this end, we claim that all of those results validate previous analysis in subsection 3.1 well. More interestingly, when $\alpha < 2$, a larger α in dPAM translates into aggravation to the bandwidth consumption, but descendent trends in figures demonstrate the basic chasing mechanism can exploit cooperation among peers more efficiently when they are equipped more resources.

Impact of Online Duration: Aiming to get a thorough understanding to P2P on demand system, we conduct experiments for different average online durations. Obviously we see a remarkable descending trend with higher $1/\mu$ in Fig.7(b) compared to that in Fig.7(a), as well as an analogous result in Fig.7(c) and Fig.7(d). The apparently depressed server workload consumption essentially comes from a smaller churn rate of nodes in P2P networks, and in turn lower probabilities for peer to become "lonely" foundlings due to the departure or failure of parents.

Impact of Buffer Size: Besides the online duration time, we also pay much attention on another important factor, i.e. the buffer size which is known to basically determine the collaboration among peers. Fig.7(a)-(d) shows that server



(a)$1/\mu = 100$, $w = 5$

(b)$1/\mu = 1000$, $w = 5$

(c)$1/\mu = 100$, $w = 10$

(d)$1/\mu = 1000$, $w = 10$

Fig. 7. Basic chasing vs. cache-and-relay

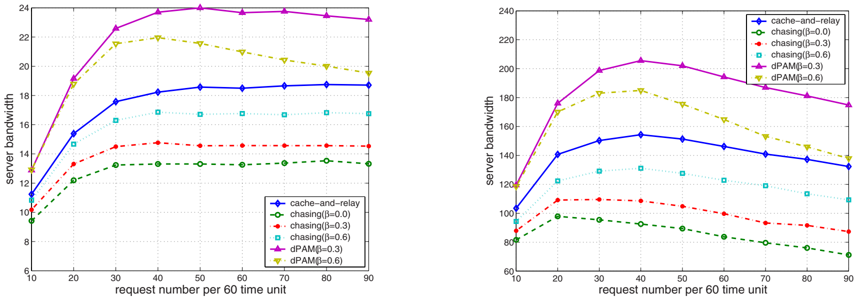
workload decreases tremendously as the buffer size increases from 5 time units to 10 time units. Note that the inflexion at 20 requests per 60 time units in Fig.7(b) disappears in Fig.7(d), which means no ”fall-after-initial-rise” exists when buffer size $w = 10$. Actually it is not surprising if we notice that the average interval between peer arrivals is 6 or 3 time units when the arrival rate is 10 or 20 requests, and accordingly a 10-unit buffer achiever much higher possibility to share the fostering burden of servers than a 5-unit buffer.

With the basic chasing mechanism, roughly only 50% server bandwidth consumption is necessary to achieve the same goal to deal with peer requests in comparison with CR scheme. These experiments have demonstrated that by efficiently taking advantages of bandwidth capacity at end users, basic chasing mechanism outperforms traditional CR scheme to provide a more scalable approach.

4.3 Performance of Advanced Chasing Mechanism

As mentioned previously, we proactively enforce more peers to prefetch contents in advanced chasing mechanism. Since the principal idea derives from the tradeoff between the server workload and transmission resilience, we attempt to concentrate on the interplay among multiple factors under various desired ratio of prefetched contents β.

Fig.8(a) and Fig.8(b) respectively depicts the server bandwidth consumption of CR, dPAM and advanced chasing mechanism over different online durations. Clearly, dPAM imposes striking workload upon server bandwidth, as we argued above, since it demands more service directly from sever when $\alpha < 2$. Although the lines describing advanced chasing mechanism always locate on top of those denoting basic chasing mechanism ($\beta = 0.0$), as a matter of factor, we conclude that the advanced chasing outperforms CR scheme and achieves higher resilience in terms of playback continuity. Fig.9 further provides an insightful comparison to the statistical ratio of prefetched peers to total population with various β. It shows that even if $\beta = 0.4$, there are more than 90% peers which have prefetched more than 2-unit contents. As a result, we improve basic chasing mechanism with the consideration that prefetching will help more peers to accommodate network fluctuation.



(a)$1/\mu = 100$, $w = 5$, $\alpha = 1.3$ (b)$1/\mu = 1000$, $w = 5$, $\alpha = 1.3$

Fig. 8. Advanced chasing vs. cache-and-relay

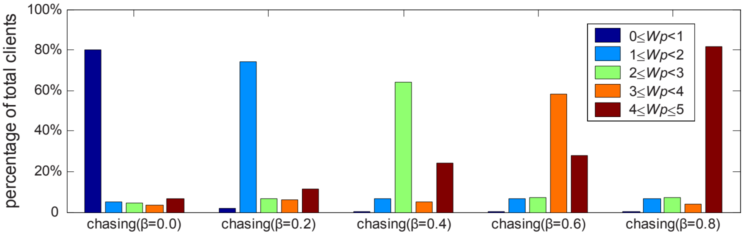


Fig. 9. Statistics of clients in prefetching content ($1/\mu = 1000$, $\lambda = 20$ per 60 time units, $w = 5$, $\alpha = 1.3$)

5 Conclusions and Future Work

In this paper, we propose an efficient streaming mechanism for scalable and resilient P2P VoD systems. We focus on how to effectively reduce the server workload in the process of fostering "lonely" peers and improve playback continuity with respect to network fluctuation. Our approach, namely basic chasing and advanced chasing mechanism, respectively helps the reduction to server bandwidth consumption by efficiently exploring surplus bandwidth for more collaboration, and offers higher resilience by proactively prefetching contents without loss of the gain to workload reduction.

The positive feedback from simulation experiments encourages us to continue our work in detailed analysis to both basic and advanced chasing mechanism, comprising the quantitative impact of buffer size, maximum downloading bandwidth, as well as prefetching amount. Our goal is to develop a practical on demand video streaming system and investigate the fundamental performance properties.

References

1. Y. Cui, B.C. Li, and K. Nahrstedt. oStream: asynchronous streaming multicast in application-layer overlay networks. *IEEE JSAC*, Vol.22, No.1, January 2004.
2. S. Jin, and A. Bestavros. OSMOSIS: scalable delivery of real-time streaming media in ad-hoc overlay networks. In *Proceedings of IEEE ICDCS'03 Workshop on Data Distribution in Real-Time Systems*, May 2003.
3. A. Sharma, A. Bestavros, and I. Matta. dPAM: a distributed prefetching protocol for scalable asynchronous multicast in p2p systems. In *Proceedings of the IEEE INFOCOM*, March 2005.
4. D.L. Eager, M.K. Vernon, and J. Zahorjan. Bandwidth skimming: a technique for cost-effective video-on-demand. In *Proceedings of SPIE MMCN*, January 2000.
5. Yang-Hua Chu, Sanjay G. Rao, and Hui Zhang. A case for end system multicast. In *Proceedings of ACM SIGMETRICS*, June 2000.
6. X. Zhang, J. Liu, B. Li, and T.-SP Yum. CoolStreaming/DONet: a data- driven overlay network for live media streaming. In *Proceedings of IEEE INFOCOM*, March 2005.
7. M. Zhang, J.-G. Luo, L. Zhao, and S.-Q. Yang. A peer-to-peer network for live media streaming - using a push-pull approach. In *Proceedings of ACM Multimedia*, November 2005.
8. A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. In *Proceedings of ACM Multimedia*, October 1994.
9. K.A. Hua, Y. Cai, and S. Sheu. Patching: A multicast technique for true video-on-demand services. In *Proceedings of ACM Multimedia*, September 1998.
10. T. Chiueh, and C. Lu. A periodic broadcasting approach to video-on-demand service. In *Proceedings of SPIE MMCN*, January 1995.
11. D.L. Eager, M.K. Vernon, and J. Zahorjan. Optimal and efficient merging schedules for video-on-demand servers. In *Proceedings of ACM Multimedia*, November 1999.
12. Y. Wang, Z.-L. Zhang, D. Du, and D. Su. A network conscious approach to end-to-end video delivery over wide area networks using proxy servers. In *Proceedings of IEEE INFOCOM*, April 1998.
13. Akamai, http://www.akamai.com/.

A Practical Approach to SIP, QoS and AAA Integration

Michael Stier, Emanuel Eick, and Eckhart Koerner

University of Applied Sciences Mannheim,
Institute for Software Engineering and Computer Networks,
Faculty of Information Technology,
Windeckstr. 110, 68163 Mannheim, Germany
{m.stier, e.eick, e.koerner}@hs-mannheim.de

Abstract. In this paper, we present an overall architecture to integrate SIP-based session management with scalable QoS features and AAA functionality. The QoS features include a bandwidth reservation in the access network based on the Next Steps in Signaling (NSIS) architecture. Furthermore, we employ the DiffServ compliant Priority Promotion Scheme (PPS) which is a packet probing scheme to verify end-to-end premium bandwidth availability. To avoid misuse of the QoS features AAA functionality is added that binds the QoS usage with the specific SIP sessions. The AAA functionality is achieved by applying a combination of the widespread RADIUS protocol with the Common Open Policy Service (COPS). As a proof of concept our architecture has been implemented and performance tested. The strength of our approach is that it keeps away complexity from the carrier networks while providing QoS with access control in a scalable fashion.

Keywords: SIP, QoS, AAA, real-time interactive services, bandwidth probing.

1 Introduction

Usage of real-time voice and video services over fixed and wireless access networks to the Internet is steadily increasing while Quality of Service (QoS) features for such services are still absent. For instance, on a DSL link a Voice over IP call may still be easily disturbed by competing data traf120fic. The rollout of existing QoS architectures is especially not progressing into the access networks as solutions for admission control to QoS features are still too complex. In this paper, we present a novel architecture that combines advanced and yet simple QoS techniques with proven AAA technologies. This combination is then integrated with SIP-based session management.

Our approach to QoS is twofold. In the access networks, we suggest bandwidth reservations for real-time voice and video where the QoS requirements are signalled through the new Next Steps in Signaling protocols. As the associated streams should be transported as premium traffic in the carrier networks we additionally propose an end system-based bandwidth probing scheme that verifies the availability of end-to-end premium bandwidth. As the best candidate for this purpose we have identified the Priority Promotion Scheme, short PPS, which is DiffServ compliant.

Radius is the well-established and most-widespread protocol to provide AAA functionality for Internet access. Consequently, we take it as the basis to control access to QoS features. It can be supported by COPS to enforce a binding between the SIP-based session management and the QoS features.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 654 – 665, 2006.

The next chapter introduces the fundamentals of SIP, PPS, NSIS and the relevant AAA architectures and protocols. From there, we derive our architecture which is presented in chapter 3. In chapter 4, we demonstrate how our architecture was implemented with open source software. The behaviour of the resulting environment is evaluated in chapter 5. We eventually draw our conclusions in chapter 6.

2 Related Work

In this chapter, we introduce all the ingredients needed to design our architecture, namely SIP, PPS, NSIS and AAA.

2.1 Session Management with SIP

The Session Initiation Protocol (SIP) [1] is a multimedia session management protocol. In conjunction with the Session Description Protocol (SDP) [2] it is the signalling protocol of choice for real-time interactive applications, including in particular Voice over IP (VoIP).

2.2 Priority Promotion Scheme

With the Priority Promotion Scheme (PPS) [3, 4] session-based on-demand Quality of Service (QoS) can be provided end-to-end in a scalable fashion. PPS uses probing packets that are sent by end systems before the actual data traffic in order to estimate the present state of a network path. The receiver of probing packets can estimate the quality of a path on the basis of packet loss or jitter. PPS is also DiffServ [5]compliant. In a Diffserv network, a classification of traffic into differently prioritized classes is made. By means of traffic conditioning all incoming packets on a router are treated according to their traffic class. For policing, probing packets are distinguished from other packets by a different Diffserv Code Point (DSCP). For instance, IP premium traffic may be marked with the DSCP value of 46 for Expedited Forwarding Per-Hop Behaviour (EF PHB) and the associated probing packets may be marked with a DSCP of 47. On the egress interface of a router a maximum rate of

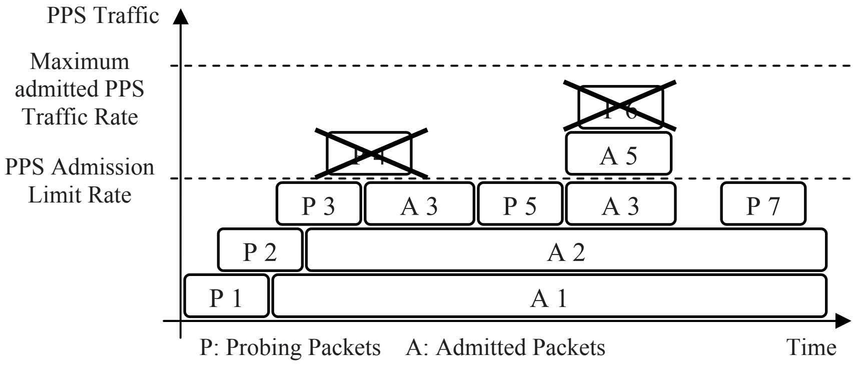


Fig. 1. Principle of PPS

PPS traffic,i.e. sum of probing and IP premium traffic, is configured (see fig. 1). Below this maximum a threshold for PPS admission control is defined. Whenever this threshold is exceeded, probing packets are dropped while IP premium traffic is still forwarded up to the configured PPS maximum rate. This behaviour guarantees bandwidth for all admitted flows while flows experiencing packet loss or considerable jitter in the probing phase are rejected.

2.3 QoS Signalling with NSIS

The IETF working group Next Steps in Signalling (NSIS) is working on a new flow signalling framework [6, 7]. NSIS is based on a two layer paradigm (see fig. 2). The lower layer is the NSIS Transport Layer Protocol (NTLP) [8] which resides on top of standard transport protocols. NTLP is responsible for delivering signalling messages from a flow source to a flow destination. The higher layer is the application-specific signalling layer with various NSIS Signalling Layer Protocols (NSLPs). For instance, NSLPs exist for NAT/firewall configuration and in particular for resource reservation signalling – the QoS NSLP [9].

At least the messaging layer must be provided for NSIS signalling on all involved network entities. The signalling sequence for QoS NSLP is depicted in fig. 3. The initiating entity is called QoS NSIS initiator (QNI). A QNI request passes one or more intermediate entities (QNE) along the NSIS path and finally terminates at the QoS NSIS responder (QNR). The request will be processed by the responder's client layer and the response travels along the reverse path to the QNI.

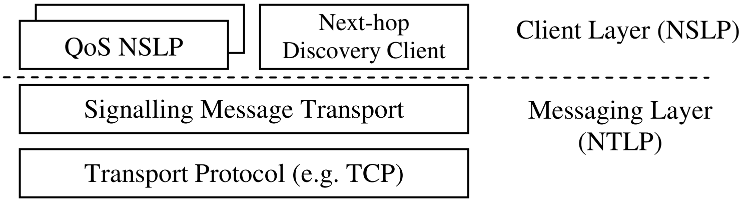

QoS NSLP
Next-hop Discovery Client
Client Layer (NSLP)
Signalling Message Transport
Transport Protocol (e.g. TCP)
Messaging Layer (NTLP)

Fig. 2. NSIS architecture

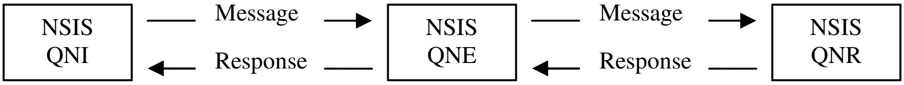


Fig. 3. NSIS entities for QoS signalling

2.4 Authentication, Authorization and Accounting (AAA)

A major interest of a service provider is that specific service features like QoS can only be used by subscribed customers. Therefore, it is important to authenticate users and authorize the usage of value added services. AAA servers are a solution for this problem. For the examination of user credentials the Remote Authentication Dial In User Service (RADIUS) [10] is the most widely deployed solution. RADIUS is characterized by its simplicity and its versatile possibilities. Although with Diameter [11] a successor is already standardized, there is no significant Diameter deployment yet.

It is very difficult to manage all entities in an ISP's infrastructure consistently, especially when these elements cooperate to provide a common service feature like QoS. Policy servers can be used to manage an administrative domain from a centralized point. To support such a policy concept the Common Open Policy Service (COPS) [12] was developed. COPS is a classical query/response protocol for exchanging policy elements between a policy server and its clients. COPS needs a so-called Policy Enforcement Point (PEP) which interprets the policy elements, received from the policy server's Policy Decision Point (PDP) [13]. The PEP has to obey the policy server's decision when performing the requested service.

3 Architecture

In the previous chapter, we have introduced several components that can be combined into a system for real-time interactive services. In our approach, the goal was to design a system which meets the requirements of an ISP. At the same time, the provisioning of QoS as an added value and its protection against misuse were of primordial importance.

The QoS part of our system takes into account the traffic situation in the Internet. The bandwidth in the access networks (e.g. mobile networks, xDSL) is scarce while the core networks are highly overprovisioned. It is expected that this situation will prevail for many years to come. With respect to QoS for real-time interactive services, this situation is problematic. For instance, on an ADSL access line a VoIP call can easily be disturbed by simultaneous data traffic from p2p applications or web downloads. The core network normally has sufficient bandwidth to transport both streams, but usually does not provide expedited forwarding for real-time interactive traffic.

Therefore, our architecture supports an explicit bi-directional reservation of bandwidth for real-time interactive services from the end systems to the network access router of the ISP. For that purpose, we adopt the QoS NSLP as described in section 2.3. Furthermore, to enable expedited forwarding of packets in DiffServ enabled core networks we employ the PPS scheme as described in section 2.2 after a successful reservation of access bandwidth has been made (see fig. 4).

As the QoS signalling is initiated by end users, there are potential security vulnerabilities. A malicious user could, for example, forge QoS signalling messages in order to gain access to prioritized service classes. The security requirements can be met by the assistance of the AAA components introduced in section 2.4. Concretely, our design choice was inspired by the token based models of the IETF's "Framework for Session Set-up with Media Authorization" [14]. In the so-called coupled model our most important requirement is addressed, namely the explicit binding of the QoS feature to a specific service. A token is issued by the AAA components to authorize usage of a service feature, in our case QoS, by the end user application. This token is fed back by the end user application into the QoS NSLP requests. Afterwards it is extracted by the access router and passed back to the policy server, as shown in fig. 4. The policy server acting as a Policy Decision Point (PDP) finally verifies the token's validity. The message loop which arises from this sequence gave it the name coupled

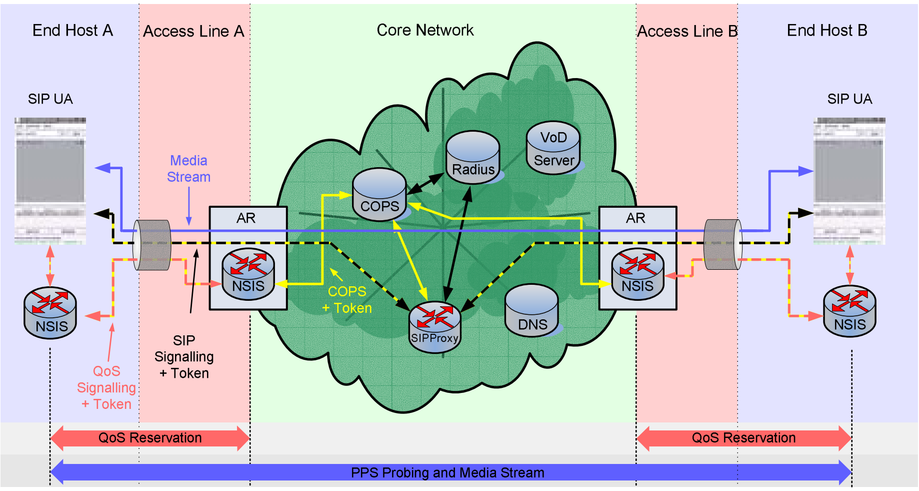


Fig. 4. Architecture integrating SIP signalling, PPS and QoS reservation with AAA

model. The model includes a Session Management Server (SMS) as an entity that provides session management services and functions as a Policy Enforcement Point (PEP). In our architecture, the SMS is a SIP Proxy.

We have integrated the coupled model with the SIP architecture according to the "Private Session Initiation Protocol (SIP) Extensions for Media Authorization" [15]. The Token is transported in an additional P-Media-Authorization SIP header field. For the token structure we adopted the IETF's "Session Authorization Policy Element" specification [16]. The token is packaged as a session identifier into the recommended policy element structure. It associates the QoS reservation with the SIP service preventing misuse of an established QoS reservation for other services.

The SIP call setup also needs to be integrated with the PPS procedure. This has been done in accordance with the specification for the "Integration of Resource Management and Session Initiation Protocol (SIP)" [17]. According to the scientific research on probe-based admission control the recommended duration of the probing phase is about one second [18]. The probing packets are sent bi-directionally between SIP User Agents (UAs). Frequency and size of the probing packets depend on the negotiated codecs. The information about available codecs is exchanged in the media description header field of the Session Description Protocol (SDP) [2] as part of SIP session setup messages prior to the probing. After the probing has finished each client analyses the received probing packets. If there are lost or truncated packets one can draw the conclusion that the network cannot provide the required end-to-end bandwidth. Otherwise, if the packet loss is zero the call setup can continue.

The SIP UA of the caller signals a busy line (all trunks busy) if the NSIS bandwidth reservation fails or the PPS function detects an insufficient end-to-end network path. The user is hence saved from a call with an inadequate quality. On the other side, if the NSIS reservation was successful and the PPS function confirms the availability of premium resources the call will be established. The negotiated media streams are established and marked with the DSCP value of 46 for expedited forwarding.

The details of the interaction between all components are illustrated by the message sequence chart in fig. 5.

The most relevant messages of the flow are:

1. The SIP User Agent Client (SIP-UAC) wants to gain access to a distinct service. It initially sends an INVITE message to the SIP proxy which challenges the client to authenticate itself with a digest authentication message [19]. The session description additionally contains the media type that is supposed to receive the QoS treatment.
2. The SIP proxy authenticates the client for the SIP service and forwards the digest authentication message to the policy server.
3. The policy server creates a request message including the digest authentication message and forwards it to the RADIUS Server.
4. The RADIUS Server validates the digest authentication message and creates a response containing the QoS status. This message is sent back to the policy server.
5. The policy server creates a token wrapping the QoS status. This token is returned to the SIP proxy.
6. Steps 4 and 5 are repeated to generate a second token for this session. After this step the authentication and authorization phase is finished.
7. The SIP proxy forwards the first token to the callee (SIP-UAS).
8. The second token is forwarded to the caller (SIP-UAC).
9. The SIP user agent is requesting resources. It creates a RESERVE message containing the token that is transferred through the UAC-QNI to the access router (AR-1 QNR) performing the QoS reservation.
10. The access router extracts the token and forwards it in a POLICY REQUEST message to the policy server.
11. The policy server searches its internal database using the token as a key. If a matching entry can be found, the stored credentials are checked and a decision is made whether the token is still valid. At this stage the loop of the token is closed.
12. If the token was successfully validated, the QoS information is packed into a positive decision message. It is sent to the entity requesting the QoS status. The policy server removes the database entry of this token so that a re-use of this token becomes impossible.
13. The AR interprets the decision message and performs the QoS reservation.
14. The decision whether a reservation was made is sent back to the requesting entity.
15. The steps 9 to 14 are repeated simultaneously on the side of the callee (SIP-UAS) with its token and access router (AR-2 QNR).
16. After the QoS state is established the probing phase begins.
17. The probing phase ends. In case of an affirmative response the user agents are synchronizing themselves with a SIP UPDATE message. The RINGING message follows. In case an insufficient network quality is detected the session will be cancelled.
18. The subsequent messages are the same as in a usual SIP call flow.

This message flow ensures that the QoS support is only granted to a valid subscriber of the SIP service. By today’s standards it is impossible for an attacker to manipulate or reuse the tokens without the detection of the fraud by the policy server.

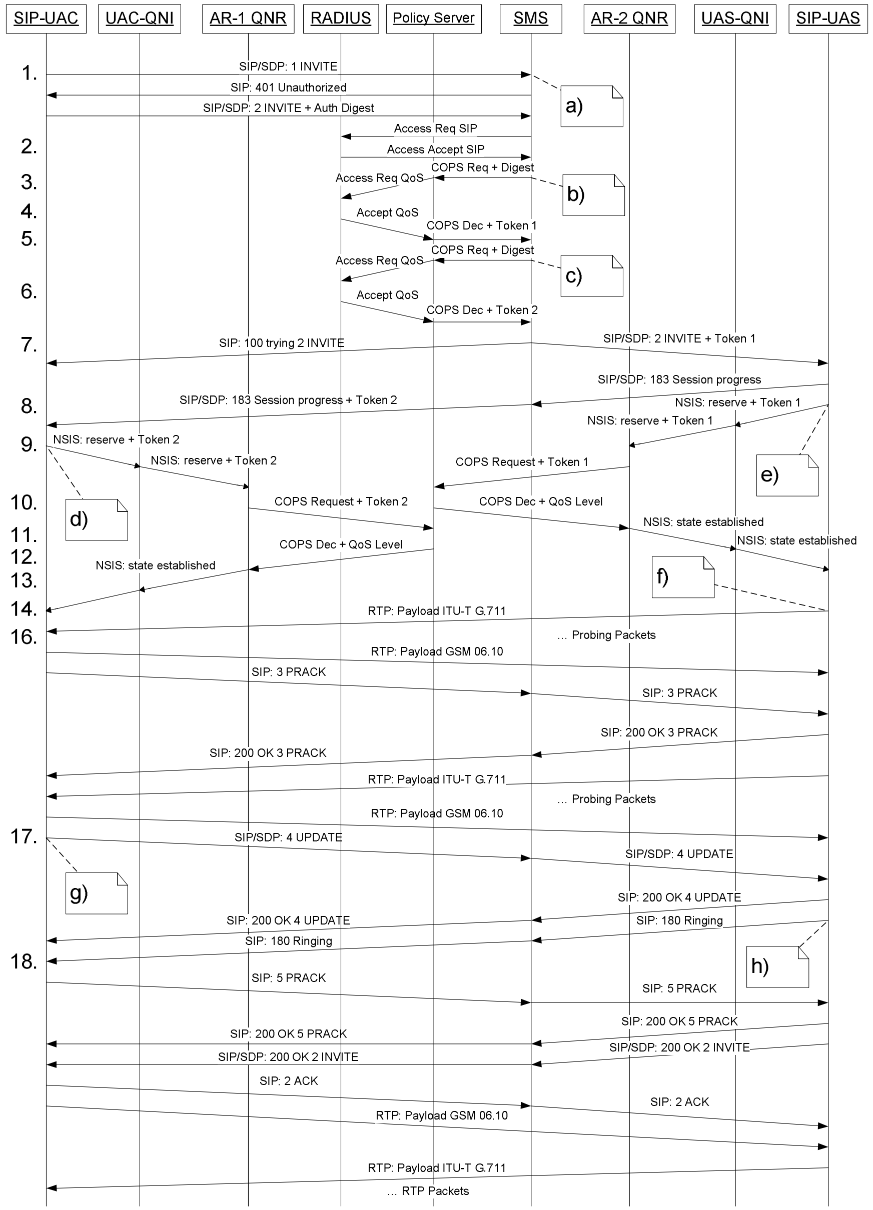


Fig. 5. SIP Call flow with PPS, QoS Signalling and AAA Coupled Model

Not shown in the figure are the REFRESH messages during the established call and the termination of the call. Due to the soft-state nature of the NSIS design each established QoS reservation has to be refreshed at an average of every 60 seconds. Otherwise, the NSIS stack has to time out the QoS reservation. If a user hangs up, the SIP session will be finished and the QoS reservation will be torn down.

4 Implementation

The presented architecture was implemented for validation and demonstration. We have realized an entire VoIP system consisting of SIP UAs, a SIP proxy, a RADIUS server, a policy server and the NSIS stack on each box. The components were distributed over 4 boxes. Table 1 shows the configuration of each box.

Table 1. Configuration of the VoIP system

	Box 1	Box 2	Box 3	Box 4
Tasks	SIP-UA 1 QNI 1	SIP-UA 2 QNI 2	SIP Proxy, AR 1, QNR 1, RADIUS, policy server, DNS	AR 2, QNR 2
OS	Linux 2.6.10	Linux 2.6.12	Linux 2.4.9	Linux 2.6.10
Processor	Mobile P3 1GHz, 256MB	M715 1,5GHz, 512MB	Mobile P3 1GHz, 384MB	PII 800MHz 256MB

Some of the components and their implementation are described subsequently.

Sip User Agent

The starting point for our extensions was the open source "SIP Communicator" user agent from the java.net project [20] which is developed at the Université Louis Pasteur in Strasbourg. First of all, we integrated the PPS procedure into this software UA. Every UA also needs one NSIS entity as initiator of the QoS signalling path. Since there is no NSIS implementation in the product stage we reused an implementation of the NSIS-like Cross Application Signalling Protocol (CASP) from University of Goettingen [21]. CASP is a predecessor of NSIS that is compliant to the requirements specified by the NSIS working group. The replacement of the CASP components with future QoS NSLP/GIST modules that are currently developed at the University of Goettingen should be straightforward. The CASP modules have been written in C. They are integrated into the Java client via a Java/C Wrapper [22]. The AAA functionality of the SIP communicator is doing the forwarding of the received token to the QoS NSIS initiator.

Network Access Router

One NSIS entity for path termination is deployed on each network access router. At the signalling layer we are using the QoS Client specified in [23] which has been implemented by the University of Goettingen [24]. Associated to this module is a COPS client, which initiates the verification of the token by the policy server. The token is extracted from the POLICY_DATA object in the CASP RESERVE message. The PPS probing procedure requires a suitable policer. This policer which is currently developed at the University of Goettingen will be integrated into the environment in the near future. It is based on the hierarchical token bucket packet scheduler available in most LINUX kernels [25].

SIP Proxy and AAA

The widely used SIP Express Router (SER) from Iptel has been installed as a SIP proxy running in version 0.9.3 on LINUX [26]. We have additionally installed the SER

RADIUS module including the RADIUS client which provides the interface for the interaction with the RADIUS server [27]. Furthermore, we developed a COPS client module for the SER that provides the interaction with the policy server for the authorization of QoS usage. The DNS holds the required SRV records for the call domain.

5 Evaluation

Based on the demonstrator the performance of the system has been studied. First of all, we discuss the timing behavior of the call flow from section 3. Figure 6 is showing the packet numbers over time beginning with the initial SIP INVITE message from the UAC right up to the RINGING message originating from the UAS.

Relevant points in time are indicated by the markers a) to h) referring to the call flow in figure 5. Table 2 provides details for each marker.

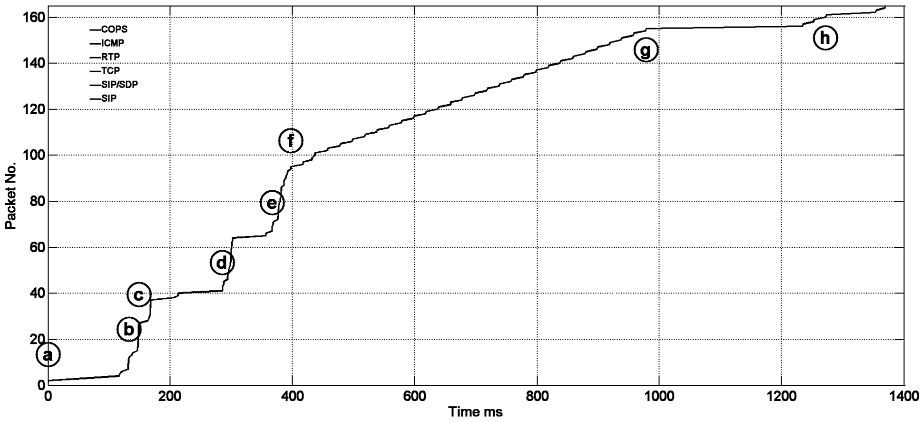


Fig. 6. Time table of SIP call flow from INVITE until RINGING message

Table 2. Time table of markers in the SIP call flow from INVITE until RINGING message

Marker	Packet No.	Elapsed Time	Description
a)	1	0 ms	INVITE
b)	12	132 ms	COPS request for token 1
c)	27	149 ms	COPS request for token 2
d)	41	286 ms	UAC QoS reservation
e)	67	367 ms	UAS QoS reservation
f)	94	397 ms	First PPS probing packet
g)	156	981 ms	Last PPS probing packet
h)	160	1272 ms	RINGING

Nearly 60% of the packets are sent in the first 400ms. One reason for this large number of packets is that the NSIS QoS signalling uses non-persistent TCP transport.The longest period of time (600ms) is needed for the probing phase marked from f)until g). It can be seen that the Java SIP UA responds relatively slow. For example, the UA needs 250ms processing time to reply with an UPDATE message after the probing phase. This test was done on box 1 to box 4 as listed in table 1. The overall time of about 1300ms for call setup is fully acceptable.

Special attention should be paid to the server components. First, we look at the RADIUS server. We wanted to know how long it takes to do a digest authentication in contrast to a simple username/password authentication. Therefore, we have done performance tests to identify the processing time for one digest authentication in a test series comprising from 10 to 100.000 requests. With rising number of requests the processing time converges in case of digest authentication to 1677μs and in case of username/password authentication to 1624μs. Thus, one can conclude that digest authentication requires only about 3 percent more processing time. This test was done on a Pentium M715 box with 512 MB Ram and Linux OS 2.6.12.

In another analysis we studied the required extra time for the AAA functionality, especially for the token-based authorization. We measured the SER processing time, i.e. the overall time needed from creating a COPS request in the PEP until the COPS decision is interpreted. In detail, the SER processing time includes the RADIUS time, the time the policy server needs to send a RADIUS access request and receive the corresponding answer and the COPS processing time, which is the time span for creating a valid COPS decision message containing the token. The plot in fig. 7 is showing the timing for each benchmark cycle on a P4 3.2GHz Linux OS 2.6.11 machine.

The mean time of 11ms to perform the authorization is satisfactory. It is a fair tradeoff considering the security gained against misuse of the QoS feature. The time for the verification of the token by the policy server can be neglected, as this task takes less than 4ms.

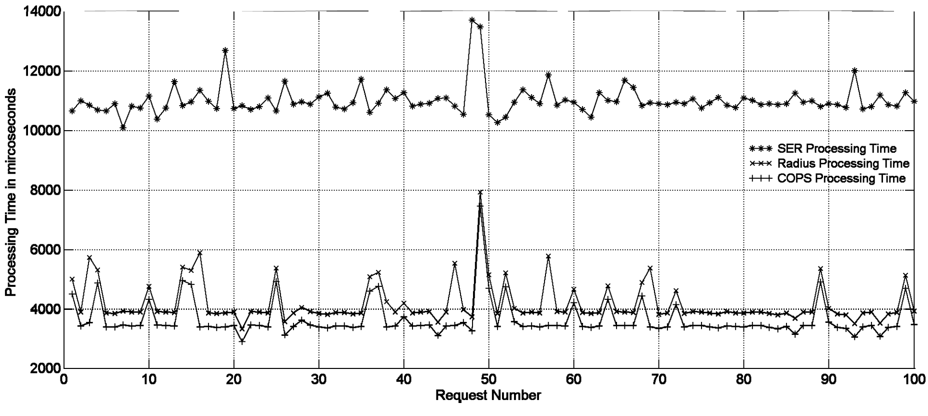


Fig. 7. Processing time for user authorization and for issuing the token

6 Conclusions

We have presented a comprehensive architecture to integrate SIP, QoS and AAA. Our architecture provides a QoS reservation on the access link that the SIP UAs initiate with NSIS. The state of the end-to-end network path is probed with PPS. PPS adheres to DiffServ. AAA functions prevent misuse of the QoS support. Only users possessing a validated token can benefit from QoS. A prototype of this architecture was implemented to verify the concepts.

PPS has been studied scientifically before. We have designed the first practical application of PPS in the context of multimedia session management. As an end-system based admission control scheme it perfectly fits the Internet philosophy that puts the complexity into the end systems. As such, PPS is also considered in the Multiparty Multimedia Session Control Working Group (mmusic) [28] where end-system based admission control is on the charter.

We have used a pairing of COPS and Radius to support AAA. When Diameter becomes more widespread in the future our architecture can easily be mapped onto a Diameter QoS application.

Finally, the presented architecture is applicable to session management in general, including in particular the Real Time Streaming Protocol (RTSP) [29]. In the latter case, one will only have to deal with unidirectional streams from the streaming server to the media players which will make NSIS signalling and PPS probing even less costly.

Acknowledgements

The work presented in this paper has been performed as part of the project GOSSIP which is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 17 19 X 04. We would especially like to acknowledge many fruitful discussions with our GOSSIP project partners at T-Systems, in particular Mr. Rüdiger Geib who drew our attention to PPS and NSIS. We also thank Xiaoming Fu and Ingo Juchem from University of Goettingen for the support related to the NSIS stack.

References

1. Rosenberg, J. et al.: SIP: Session Initiation Protocol. IETF RFC 3261, June 2002
2. Handley, M., Jacobson, V.: SDP: Session Description Protocol. IETF RFC 2327, April 1998
3. Mori, S., Kawarasaki, Y., Kataoka, H. and Morita, N.: Priority Promotion Scheme (PPS) - An Autonomous and Distributed Admission Control for End-to-end Quality of Service for Interactive Multimedia Services, NTT Technical Review Online, October 2004, http://www.ntt.co.jp/tr/0410/special.html
4. Morita, N., Karlsson, G.: Framework of Priority Promotion Scheme. IETF Internet Draft draft-morita-tsvwg-pps-01, October 2003
5. Blake, S. et al.: An Architecture for Differentiated Services. IETF RFC 2475, Dec. 1998
6. Hancock, R. et al.: Next Steps in Signaling (NSIS): Framework. IETF RFC 4080, June 2005

7. Fu, X. et al.: NSIS: A New Extensible IP Signaling Protocol Suite. In: IEEE Communications Magazine, Internet Technology Series, page 133-141, IEEE, October 2005
8. Schulzrinne, H. et al.: GIST: General Internet Signaling Transport. IETF Internet Draft draft-ietf-nsis-ntlp-08, work in progress, September 2005
9. Van den Bosch, S., et al.: NSLP for Quality-of-Service signalling. IETF Internet Draft draft-ietf-nsis-qos-nslp-08, work in progress, October 2005
10. Rigney, C. et al.: Remote Authentication Dial In User Service (RADIUS). IETF RFC 2865, June 2000
11. Calhoun, P. et al.: Diameter Base Protocol. IETF RFC 3588, September 2003
12. Durham, D. et al.: The COPS (Common Open Policy Service) Protocol. IETF RFC 2748, January 2000
13. Yavatkar, R. et al.: A Framework for Policy-based Admission Control. IETF RFC 2753, January 2000.
14. Hamer, L-N. et al.: Framework for Session Set-up with Media Authorization. IETF RFC 3521, April 2003
15. Marshall, W. (ed.): Private Session Initiation Protocol (SIP) Extensions for Media Authorization. IETF RFC 3313, January 2003
16. Hamer, L-N. et al.: Session Authorization Policy Element. IETF RFC 3520, April 2003
17. Camarillo, G., Marshall, W., Rosenberg, J.: Integration of Resource Management and Session Initiation Protocol (SIP). IETF RFC 3312, October 2002
18. Más Ivars, I., Karlsson, G.: PBAC: Probe-Based Admission Control. In Proc. of QofIS 2001, vol. 2156 of LNCS, (Coimbra, Portugal), pp. 97-109, Springer, September 2001
19. Franks, J. et al.: HTTP Authentication: Basic and Digest Access Authentication. IETF RFC 2617, June 1999
20. SIP User Agent Sip-Communicator, https://sip-communicator.dev.java.net
21. Schulzrinne, H., Tschofenig, H., Fu, X., McDonald, A.: CASP - Cross-Application Signaling Protocol. IETF Internet Draft draft-schulzrinne-nsis-casp-01.txt, March 2003 http://user.informatik.uni-goettingen.de/~casp/draft-schulzrinne-nsis-casp-01.pdf
22. Simplified Wrapper and Interface Generator (SWIG), http://www.swig.org
23. Schulzrinne, H. et al.: A Quality-of-Service Resource Allocation Client for CASP. IETF Internet Draft draft-schulzrinne-nsis-casp-qos-01.txt, March 2003, http:// user.informatik. uni-goettingen.de/ ~casp/draft-schulzrinne-nsis-casp-qos-01.pdf
24. Juchem, I.: An Implementation of QoS NSLP. http://user.informatik.uni-goettingen.de/ ~qos/
25. Severa M.: Hierarchical token bucket (HTB) home. Packet Scheduler, http://www. huisetalage .nl/ sip/index.html
26. Sip Express Router (SER), Free SIP Server, http://www.iptel.org/ser
27. freeRADIUS, Open Source RADIUS Server Project, http://www.freeradius.org
28. Multiparty Multimedia Session Control (mmusic), IETF working group, http://www.ietf. org/ html.charters/mmusic-charter.html
29. Schulzrinne, H. et al.: Real Time Streaming Protocol (RTSP). IETF RFC 2326, April 1998.

Efficient Overlay Audio Conferencing

Norbert Egi, Nick Blundell, and Laurent Mathy

Computing Department, Lancaster University,
Lancaster, LA1 4WA, UK
{egi, n.blundell, laurent}@comp.lancs.ac.uk

Abstract. In this paper, we present a thorough and realistic analysis of audio conferencing over application-level multicast (ALM).

Through flexibility and ease-of-deployment, ALM is a compelling alternative group-communication technique to IP Multicast — which has yet to see wide-scale deployment in the Internet. However, proposed ALM techniques suffer from inherent latency inefficiencies, which we show, through realistic simulation and exploration of perceived quality in multi-party conversation, to be greatly problematic for the realisation of truly-scalable audio-conferencing systems over ALM.

In this work, we propose to adapt dynamically the application-level distribution structure to the conversational pattern of the audio conference. The contribution of this paper is threefold: we develop a novel perceptual quality model for multi-party audio conversations; we provide dynamic adaptation via a simple next-speaker prediction technique and we validate the proposed approach by using a large and detailed corpus of real multi-party conversations.

Keywords: ALM, VoIP, Conversation Analysis, Adaptive Conferencing.

1 Introduction

It is well known that the mouth-to-ear latency of an echo-less voice-communication channel should not exceed 300 ms in order to allow natural conversation [2] — audible echo can reduce this threshold by two orders of magnitude.

This limit is especially important for Internet VoIP applications, where the communication channel comprises non-trivial application and network latency components. With typical one-way application latencies of 60–400 ms [6] and Internet round-trip latencies of 150–200 ms [4], such VoIP applications operate with communication-channel latencies that are at or above the upper threshold of human tolerance.

In particular, this poses a problem for application-level group communication techniques, which are inherently less latency- or cost-efficient than their scarcely-deployed network-level counterpart (*i.e.* IP multicast), for example: multiple unicasting between participants cannot scale to support even modestly sized groups; standard overlay-tree flooding (*i.e.* that is performed by proposed application-level multicast (ALM) techniques) results in highly-varied node-pair latencies; and centralised reflector servers do not accommodate well groups of

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 666–677, 2006.

widely distributed membership (*i.e.* since there is no obvious place to put the reflector). A specific solution is therefore required to support group audio applications.

In [1] we proposed ALNAC, a dynamic application-level multicast (ALM) routing protocol especially designed for audio-conferencing applications, and we argued that perceptual quality of multi-party conversation could be improved by exploiting the patterns in natural conversation that allow for prediction — with a high accuracy — of who will speak next in conversation.

In this paper, we develop the preliminary work in [1] into a thorough investigation of the problem and make the following contributions. In Section 2, we give an in-depth exploration of the specific effects of communication-channel latency on multi-party conversation, leading to a novel model of perceptual quality. In Section 4, we propose, and conduct a thorough analysis of, several approaches to next-speaker prediction algorithms, using a large corpus of highly-detailed talkspurt data from actual multi-party conversation. Finally, in Section 5, we evaluate, by simulation, our ALM-based audio-conferencing proposal under conditions of realistic network latency and through using a realistic model of multi-party conversation.

2 Issues of Latency in Multi-party Conversation

In interactive scenarios, latency is a problem usually because it *cannot* be perceived: in fact, only when a source's sound is reflected (echoed) back can latency be gauged; otherwise, the listener's brain interprets what is heard or what is not heard as events that happen in real-time, for example: we would quite-happily perceive a live radio show as such despite that it may in fact have a two-minute censorship delay.

Thus, when engaged in conversation we are constantly (subconsciously) projecting times at which *responses* to our spoken *cues* should arrive; if they do not arrive within our expected time range (a range bounded by well-studied latency-tolerance threshold), we perceive that they will never arrive and repeat cues unnecessarily in an attempt to repair the conversation.

With these two considerations in mind, we can argue, therefore, that perceived quality is not simply dependent upon a communication channel's latency but upon the delay with which specific responses are heard after their cues. This observation is particularly relevant to multi-party conversation, since not only will a participant hear responses to their own cues — if they choose to speak — but they will also hear responses to the cues of other participants, and so perceptual quality of those responses will be dependent not upon the *absolute latencies* with which they are heard but upon the *difference* in absolute latencies of those responses and their cues.

We remark that not all speech acts cue a response (for example at the end of the discussion on a topic). On the other hand, one should keep in mind that conversational turns (i.e. change of speaker) are typically delimited by silence gaps of no more than 1 second [11].

2.1 Issues of Stream Synchronisation

Due to the inherent latency inefficiency of ALM techniques, there is a potential that participants of a multi-party communication system will observe highly-varied network delays between streams, which, by affecting the synchronisation of responses and their cues, will impact upon perceived quality.

To the best of our knowledge, there has been no study on the effects of stream desynchronisation on the perceived quality of multi-party conversation (*i.e.* from a listener's perspective). We therefore performed a simple listening experiment in which we desynchronised a participant channel of a recorded meeting from the ICSI meeting corpus [5]: in the experiment, we took one recorded audio channel (a channel of a participant who was engaged in conversation for a large proportion of the particular meeting segment) and shifted it by various time constants, before mixing all of the separate channels into a single audio file; a set of mixes were thus created (including the original mix without shifting). Volunteers, who had no insight into the particular transformations that was performed, were asked to categorise between those mixes that sounded "strange" and those that sounded normal.

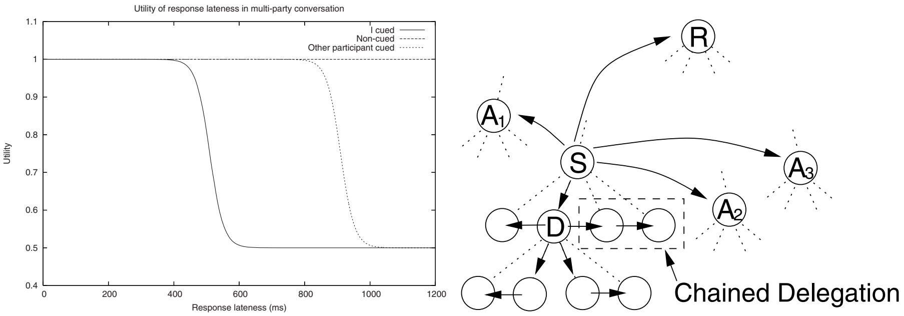
Interestingly, none of the listeners in the experiment could perceive a difference between mixes that were desynchronised by less than or equal to 1,000 ms, which indicates that we have a higher tolerance to the lateness of responses when we listen to a conversation than when we are actively engaged in it (in which case the maximum mouth-to-ear round-trip tolerance is about 600 ms).

2.2 Quality Model for Multi-party Conversation

In line with our observations on quality-perception in multi-party conversation, we propose a perceptual-quality model that is not based on channel latency, as has so far been considered in the literature, but rather on the 'lateness' of individual spoken responses with respect to their cues.

In [2], the authors proposed a simple utility function for describing the perceived quality of mouth-to-ear channel latency in which two score-levels are defined: a high score level to reflect 'very good' latency perception, and a low score level to reflect 'bad' latency perception. We base our own utility function on a similar principle but instead consider tolerance to round-trip mouth-to-ear latency, since conversation — as a two-way process — is affected by round-trip latency and since asymmetric latencies are highly-likely in ALM. In addition to the original utility function, we extend the function to distinguish between the tolerance thresholds to response lateness for a participant's own cues and for the cues of other participants.

The resulting utility function for perceptual quality is depicted in Figure 1(a). Note that, in the model, since a non-cued talkspurt cannot be perceived as being late, it is automatically awarded a score of 1.



(a) (b)

Fig. 1. (a) Utility function for the perceptual quality of response lateness in multi-party conversation; (b) Dynamic routing of ALNAC through the process of delegation

3 ALNAC: A Dynamic Overlay Routing Protocol

ALNAC (Application-Level Network Audio-Conferencing routing protocol) [1] is a light-weight ALM routing protocol, designed especially to optimise audio-packet delivery for those audio-conference participants who are most sensitive to communication-channel latency (*i.e.* those who are currently engaged in conversation), whilst minimising the impact of such optimisation on members that are least sensitive to communication-channel latency (*i.e.* those members that take only a listening role in the current conversation).

More precisely, ALNAC operates over an ALM tree structure. ALNAC adapts a basic flooding technique whereby a speaker sends audio samples to the tree root and to its children (for forwarding in their respective sub-trees). The adaptation is that a speaker will send audio samples, in addition to the root, directly to a set of predicted next speakers who are identified as highly likely active participants in the current conversation by a prediction algorithm. On the other hand, because the out-degree of a node (i.e. the maximum number of forwarding the node will do) can be limited due to bandwidth constraints, some of the speaker's children on the ALM tree may have to be *deprived* from receiving the audio samples from the speaker directly. To ensure that all samples are eventually flooded to all nodes in the tree, a speaker will *delegate* the responsibility for supplying the deprived nodes among the nodes to whom it *is* sending directly. Note however, that delegation can be recursive (i.e. a supplier can further delegate). Figure 1(b) illustrates the audio sample distribution process in ALNAC.

We therefore see that, in essence, ALNAC builds a dynamic overlay over an ALM tree (as opposed to adapting the tree to conversation changes).

Note that in [1], a very primitive, static prediction algorithm was proposed. A more efficient and adaptive algorithm is described in the following section.

4 Next-Speaker Prediction

From Section 3, it should be clear that the ability of ALNAC to identify the participants that are actively engaged in conversation is critical for the effectiveness of the protocol.

In [1], through the analysis of a limited number of textual transcripts of actual conversation and packet-trace files of an audio-conferencing application, we showed that in natural, multi-party conversation there is a high correlation between those participants who spoke recently and those who will speak next; the explanation for this result lies in the relationships between conversational turns, such as *adjacency pairs* (*e.g.* questions and answers, exclamations and responses, *etc.*), which have been well-documented in the study of conversation analysis [10]. We also devised a rudimentary next-speaker prediction algorithm.

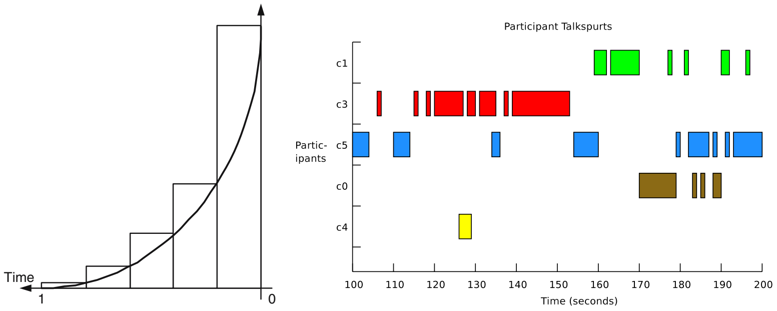
In the context of audio conferencing over application-level multicast, we define the problem of next-speaker prediction as a problem of maximising the probability that one participant of a constrained set of recent-speaking participants, which set we refer to as the *backlog*, will speak next. Thus, the role of a next-speaker prediction algorithm essentially is to create a prioritised list of participants, ranking them by their level of 'activeness' in the conversation, such that a minimum backlog may be extrapolated from the priority list to perform optimised overlay routing.

In this section, we extend our previous work on next-speaker prediction into a more-complete analysis through the incorporation of corpus data collected and processed by the ICSI meeting project [5]. The corpus comprises the data of over seventy full-length meetings of natural, multi-party conversation, featuring interactions among wide varieties of participants (*i.e.* gender, age, ethnicity, *etc.*), and was produced primarily to aid linguistical research on group conversation and interaction. The data for each meeting comprises recordings of per-participant audio and highly-detailed transcripts, painstakingly annotated per-talkspurt with timing and semantic information. Figure 2(b) shows a sample of talkspurt patterns plotted from corpus meeting.

Using only timing information of talkspurts, as is readily available with little processing overhead to participants of an audio conference, we present, due to lack of space, only our most sophisticated next-speaker prediction algorithm, that gives the best overall efficiency under the most circumstances and is heuristically derived through talkspurt analysis of the ICSI meeting corpus data; furthermore we give an evaluation of the algorithm as well.

4.1 Prediction Algorithm

The next-speaker prediction algorithm presented in this section follows a strategy of associating with each group member (*i.e.* conference participant) a *priority* which quantifies the recent conversational contribution of the participant (and thus, the participant's likely immediate future contribution). The algorithm takes as input (a description of) audio samples/packets and produces a list of participants (ordered according to their computed priorities) on detection of



(a) (b)

Fig. 2. (a) Calculation of sub-window weights. (b) A sample of talkspurt activity among participants of the ICSI meeting corpus.

turn boundaries (*i.e.* at points of speaker-change) to reflect changing levels of participation throughout the course of the meeting.

Analysis of the ICSI corpus shows that disregarding turns shorter than 800 ms alleviate the problem of falsely indicating that a particular participant is currently engaged in conversation, that is caused by unintentional talkspurts (*e.g.* environmental noises) or intentional back-channel talkspurts (*e.g.* 'hmmm').

The reader should note that each participant runs an independent instance of the next-speaker prediction algorithm and that a description of all audio packets (both received and produced by a participant) are used as algorithm input.

Our next-speaker prediction algorithm analyses both the recent turn-activity of participants and the *occupancy* (*i.e.* the cumulative duration) of their talkspurts within a recent time window: participants who produced talkspurts within the bounds of the window are awarded a priority based upon an occupancy calculation of their talkspurts, and those participants who have no talkspurts within the window are given a priority of zero (*i.e.* they may be considered not to be engaged in the current conversation).

To give weight to those talkspurts that occurred more-recently in the window (*i.e.* so that participants with more-recent talkspurts are favoured), the time window is decomposed into n sub-windows of equal length and a weight is calculated for each sub-window based on some decay function. Figure 2(a) depicts this decomposition, where sub-window weights are calculated as the area of rectangles delimited by the decay function.

4.2 Practical Considerations

Our next-speaker prediction algorithm is conceptually simple and easy to implement. However, in order to achieve effective ALNAC routing in all circumstances, we propose some simple extensions.

The next-speaker prediction algorithm simply returns a priority-ordered list of participants. Although always using the maximum allowed backlog increases ALNAC's effectiveness, but it also results in a higher protocol overhead caused by more delegation (see Section 3). In order to keep actual backlog sizes (and therefore delegation) to a minimum, while still achieving high prediction accuracy, we introduce the concept of *backlog priority threshold*, whereby prediction algorithms will only return a priority-ordered list of participants whose priority is higher than the backlog priority threshold. Obviously, a higher threshold forces the algorithms to 'hide' participants whose recent conversational contribution is 'minor'.

Since in some applications, such as tele-teaching, the turn of a participant with, say, a teaching role can be expected to last for a very long duration of time (*i.e.* with only sparse interruptions by students throughout the duration of the session) we propose that, in order to avoid unnecessary overhead of the ALNAC routing process (*i.e.* by having an unnecessary-large backlog), algorithm priorities should be re-computed periodically intra-turn with a period of M ms (*i.e.* in addition to being computed on turn boundaries), such that the previous turn of, say, a student becomes discounted after a period of time *during* the teacher's very long turn.

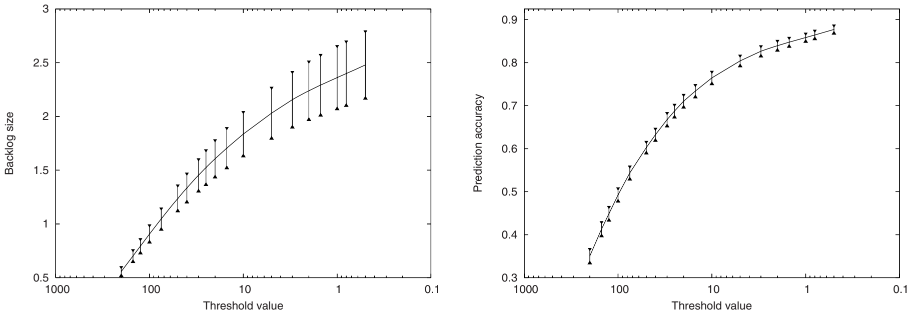
4.3 Evaluation of the Algorithm

We have defined our new next-speaker prediction algorithm that, through the observation only of recent talkspurt patterns among participants of multi-party conversation, tries to maximise the accuracy of next-speaker prediction. In this section, we evaluate the efficiency of this algorithm (*i.e.* its ability to predict the next speaker with a minimised backlog).

To understand the relationship between the backlog-priority threshold, the backlog size and, of course, prediction accuracy, we performed an analysis of the algorithm over talkspurt data in all 75 ICSI meetings of the corpus, computing the average and 95% confidence intervals for backlog size and prediction accuracy for a range of threshold values that were chosen sensibly to suit the priority mechanism of the algorithm (see Figure 3).

In Figure 4 the results of threshold analysis for our new and 'Initial' algorithms have been combined to show the prediction accuracy of each algorithm against average backlog size. Note that, the 'Initial algorithm' represents the prediction accuracy of our rudimentary algorithm from [1], in which prioritization of participants for prediction is based upon only the order of recent talkspurts.

In Figure 4, we see that our new algorithm gives improved performance over the initial algorithm; this occurs as a result of this algorithm being capable of making intelligent judgements as to whether a talkspurt is significant in prediction or not, for example: whether a talkspurt is a short burst of noise or back-channel speech (*i.e.* 'mmm-hmm', 'yeah', *etc.*) from participants who have no intent to become engaged in the current conversation. In summary, we see that a high prediction accuracy may be achieved in multi-party conversation by considering only a small backlog of previous speakers (≤ 3); this result confirms the results of our less-extensive analysis of textual transcript turn patterns and packet traces in [1].



(a) (b)

Fig. 3. Backlog-priority threshold characteristics of the new algorithm

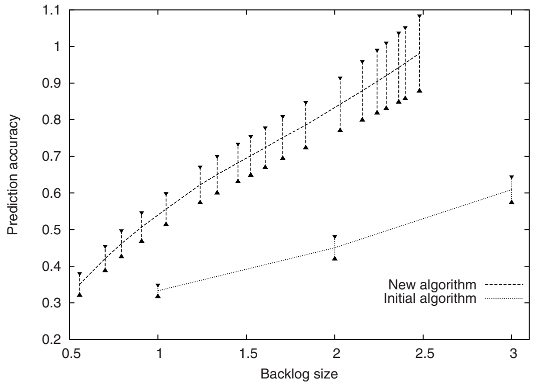


Fig. 4. Comparison of the algorithms' prediction efficiency

5 Simulations

Since existing topology-based simulators cannot simulate the realistic dynamics of Internet node-pair latencies (*i.e.* due to complexities of traffic patterns and network structuring), we implemented an event-based network simulator that uses latency matrices, populated by actual Internet latency measurements. The latency matrices were obtained for 1740 arbitrary Internet hosts from [3] and for PlanetLab [9] nodes from [12].

The goal of ALNAC is to improve the perceptual quality experienced by not *some* but *all* participants of an Internet audio conferencing system over application-level multicast, and so here a performance comparison is made among ALNAC and two other approaches, Narada and TBCP, that are representative of conventional (*i.e.* non-adaptive) per-source–tree and shared-tree application-level multicast, respectively:

- the performance of Narada [13] is important as a benchmark since, through a technique of exhaustive probing to construct a highly-optimised overlay network — albeit doing so with a high protocol overhead — the protocol is designed especially to provide low-latency, any-source group communication;
- the performance of TBCP [8] is important as a benchmark since the protocol is able quickly to build good quality shared trees with a low protocol overhead, and is thus representative of more scalable, alternative approaches to Narada.

Two additional ALM protocols used in the comparison are multiple-unicast, representative of the latency performance of network-level multicast, and a variation on multiple-unicast which is introduced here as *magic multicast*: through an exhaustive off-line search of the simulator's latency matrix, the magic-multicast ALM protocol exploits all of the best shortcut paths that improve latency over those of unicast, and is therefore representative of an ALM protocol with the best possible latency performance.

Recall that ALNAC defines only the way in which (audio) data should be flooded over a given overlay network and not the structure of the overlay network itself, and so ALNAC may run on top of any overlay network from which distribution trees may be inferred. In this case ALNAC is run on top of a (shared) TBCP tree to demonstrate how such shared-tree overlay networks can benefit from dynamic flooding whilst maintaining properties of good scalability — indeed, ALNAC may be run over, say, Narada, but the benefit of doing so would be less.

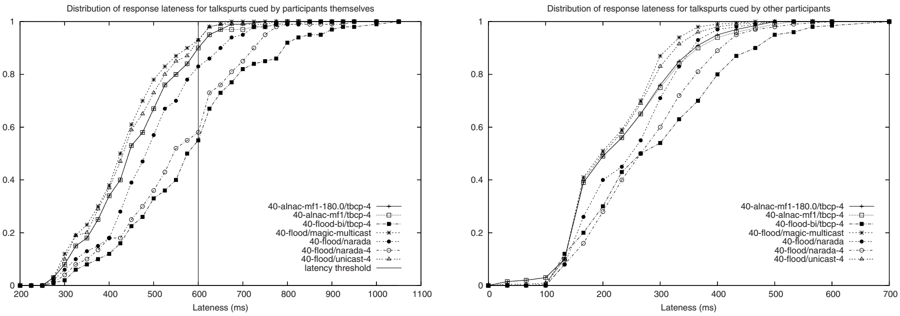
In the simulations ALNAC was run using our new next-speaker prediction algorithm introduced in Section 4.

5.1 Results

Figure 5 plot conversational response lateness performance of the ALM protocols for simulations of an audio conference among 40 nodes. The simulations were run using reconstructed conversation among participants of four of the largest (participant-wise) transcripts from the ICSI meeting corpus, with ten repetitions of each simulation (*i.e.* with random selection of overlay nodes from the King latency matrix [3] and random placement of ICSI meeting participants on overlay nodes). Note that, where applicable, a low maximum out-degree constraint of 4 is enforced on the overlay construction protocols (*e.g.* `narada-4`, `tbcp-4`) to be representative of a low and fair forwarding overhead for each node.

Note that, Narada was simulated both with and without a maximum out-degree constraint, since given the freedom of being unbound, the protocol is better able to optimise the overlay for latency.

Figure 5(a) shows the distribution of response lateness for conversational responses that were cued by participants themselves (*i.e.* those participants that were engaged in conversation) and Figure 5(b) shows response lateness for responses cued by other participants (*i.e.* those heard by participants who were not at that time engaged in conversation).



(a) (b)

Fig. 5. Distributions of conversational response lateness

Figure 5(a) captures the essence of ALNAC, since despite in the experiment that ALNAC was run over a shared TBCP overlay tree and with a maximum node out-degree of only 4, perceptual lateness for participants that became engaged in conversation throughout the simulated audio conference is significantly improved over that of the other conventional ALM protocols; even improving over that of Narada with an unbound maximum out-degree (`flood/narada`), which had a tendency to burden centrally-located nodes with a large out-degree of 9–11; in fact, the performance of ALNAC is very close to that of the ideal protocols `flood/unicast` and `flood/magic-multicast`.

This is actually as expected, since ALNAC exploits the predictability of conversational patterns in order to deliver — through a process of semi–tree-circumvention — audio data with minimal latency to those participants who are currently engaged in conversation, whereas the other ALM protocols, during the overlay-network construction phase, try to strike a balance of optimisation for the whole group with the result that, when the time comes for them to communicate, a significant proportion of participant-pairs will be unable to tolerate the excessive communication-channel latency.

In Figure 5(a), the only reason that the performance of ALNAC does not mirror exactly that of multiple-unicast and magic-multicast is that, although highly accurate due to the natural ordering of turn-taking that occurs in multiparty conversation, next-speaker prediction is not infallible and nor is the identification of cues and responses that are used to measure performance of the protocols: next-speaker prediction often falters during the occasional 'hot spots' that occur in multiparty conversation.

Figure 5(b) is very interesting, since by capturing the perceptual quality experienced by participants that listened to the conversation of other participants without themselves being engaged in conversation at that particular time, it is complementary to Figure 5(a), and thus completes the picture of perceptual quality experienced by *all* participants of the simulated audio conferences. As argued in Section 2, a non-engaged participant (*i.e.* as opposed to a participant who their self is issuing cues) will perceive a response to be late not by the absolute latency with which it is received but by its relative lateness with respect to the time that the cue was received; this feature of perception is captured in Figure 5(b) by the fact that, despite there being some variation in the degrees of lateness experienced by participants under the various protocols, none of the responses were actually significantly late as to impair perceptual quality of the participants, for any of the protocols; the reason that these distributions have consistently smaller ranges than those of Figure 5(a) is because, from the perspective of a non-engaged participant, cues and responses undergo the same or similar application delays (*i.e.* packetisation, decoding, and buffering), which are effectively cancelled between two incoming streams, such that, and unlike for participants who are engaged in conversation, perceptual lateness of non-engaged participants is actually a reflection only of the difference in the respective network delays incurred by incoming streams. The curves of this figure thus reflect the distributions of path lengths of the overlay networks build by the various ALM protocols.

An interesting observation to make of Figure 5(b) is that for non-engaged participants ALNAC actually improves perceptual lateness over TBCP with conventional flooding yet both use the same shared overlay tree; this can be explained by the fact that ALNAC nodes will make full use of their maximum forwarding capacity to disseminate data on the tree, for example: with conventional shared-tree flooding, a leaf node will begin dissemination only through a single node (*i.e.* its parent or the tree's root); whereas an ALNAC node will also or alternatively begin dissemination through a number of nodes that host those participants who are predicted to be engaged in conversation at that particular time.

6 Conclusions

In this paper, we have presented a novel and thorough investigation of two properties of multi-party conversation that are highly important in the realisation of VoIP applications over ALM: (i) the effects of communication-channel latency on quality perception in multi-party conversation; and (ii) the problem of next-speaker prediction in multi-party conversation (*i.e.* which participants are at current most-sensitive to communication-channel latency). The two main contributions, namely our the quality model for multi-party conversation and our efficient next-speaker prediction algorithm, although developed in the context of our work on ALNAC, are readily applicable in the wider context of audio-conferencing systems. Indeed, they can, for instance, be used to evaluate and guide the operation of related proposals such as ACTIVE[7] (a proposal based on the strategy of shaping the ALM tree so that active speakers are near the root).

We have also presented the ALNAC protocol and conducted simulations that model, realistically, characteristics both of the network and of multi-party conversation.

Based on our analysis and simulations, we conclude that in order to support truly-scalable audio conferencing over ALM, an ALM routing protocol *must* be reactive to the conversational patterns of participants, such that perceived quality may be improved for not just *some* of the participants (*i.e.* by fortune of their location in the overlay tree(s)) but for *all* participants. The ALNAC protocol, including its next-speaker prediction algorithm, was shown to be a scalable, elegant and general solution to this problem, capable of efficiently supporting both meeting-type and orator-type audio conference applications.

References

1. Blundell, N., Mathy, L.: Minimising *Perceived* Latency in Audio-Conferencing Systems over Application-Level Multicast. In Proceedings of MIPS 2004, Grenoble, France, Nov 2004.
2. Boutremans, C.,Le Boudec, J.-Y.: Adaptive Delay Aware Error Control For Internet Telephony. In Proceedings of the 2nd IP-Telephony Workshop, New York, April 2001.
3. Gil, T. M., Kaashoek, F., Li, J., Morris, R., Stribling, J.: King Dataset. http://www.pdos.lcs.mit.edu/p2psim/kingdata/, August 2004.
4. Gummadi, K.P., Gummadi, R., Gribble, S. D., et al.: The Impact of DHT Routing Geometry on Resilience and Proximity. In Proceedings of the ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.
5. Janin, A., Ang, J., Bhagat, S., et al.: The ICSI Meeting Project: Resources and Research. Proceedings of NIST ICASSP 2004 Meeting Recognition Workshop, Montreal, May 2004.
6. Jiang, W., Koguchi, K., Schulzrinne, H.: QoS Evaluation of VoIP End-Points. Proceedings of IEEE International Conference on Communications (ICC 2003), Anchorage, Alaska, May 2003.
7. Liu, L., Zimmermann, R.: ACTIVE: Adaptive Low-Latency Peer-to-Peer Streaming. In Proceedings of the Twelfth Annual Multimedia Computing and Networking (MMCN '05), San Jose, California, January 2005.
8. Mathy, L., Canonico, R., Hutchison, D.: An Overlay Tree Building Control Protocol. Proc. of Intl. workshop on Networked Group Communication (NGC), Nov 2001. 76–87
9. PlanetLab. http://www.planet-lab.org.
10. Lectures on Conversation. Blackwell, Oxford, UK, 1992.
11. Schmitz, U.: Eloquent Silence. Linguistik-Server Essen (LINSE), 1994.
12. Stribling J.: PlanetLab all pairs ping data. http://pdos.csail.mit.edu/~strib/pl_app/.
13. Chu Y.-H., Rao S. G.,Zhang H.: A Case for End-System Multicast. In Proceedings of ACM SIGMETRICS 2000, Santa Clara, California, US, June 2000.

On the Stability of End-Point-Based Multimedia Streaming*

György Dán, Viktória Fodor, and Gunnar Karlsson

Department of Signals, Sensors and Systems,
KTH, Royal Institute of Technology
{gyuri, vfodor, gk}@s3.kth.se

Abstract. In this paper we propose an analytical model of a resilient, tree-based end-node multicast streaming architecture that employs path diversity and forward error correction for improved resilience to node churns and packet losses. Using the model and via simulations we study the performance of this architecture in the presence of packet losses and dynamic node behavior. We show that the overlay can distribute data to nodes arbitrarily far away from the root of the trees as long as the loss probability is lower than a certain threshold, but the probability of packet reception suddenly drops to zero once this threshold is exceeded. The value of the threshold depends on the ratio of redundancy and on the number of the distribution trees. Using the model and simulations we show that correlated and inhomogeneous losses slightly worsen the overlay's performance. We apply the model to study the effects of dynamic node behavior and compare its results to simulations.

1 Introduction

The delivery of streaming media over end-point overlays has received much attention recently ([1, 2] and references therein). Although current commercial content delivery networks are capable of supporting many simultaneous streams, end-node-based multicast could considerably decrease the cost of large scale streaming, while being resilient to sudden surges in the client population, such as flash crowds. In an end-point-based multicast distribution system end-points are organized or organize themselves into an application layer overlay and distribute the data among themselves. The main advantages are that such a system is easy to deploy and it reduces the load of the content provider, since the distribution cost in terms of bandwidth and processing power is shared by the nodes of the overlay.

Since the success of such schemes depends on the behavior of the participating nodes, several issues have to be dealt with, such as the effects of group dynamics, stability of the system or the incentives for nodes to collaborate. Furthermore, since nodes receive data from their peer nodes only, the performance

* This work has been supported in part by E-NEXT and by the Swedish Foundation for Strategic Research under the program Affordable Wireless Services and Infrastructures.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 678–690, 2006.

of such a scheme in an error prone environment is unclear due to possible error propagation.

The first proposed architectures focused primarily on low overhead due to control traffic and on the efficiency of the data distribution. They were based on a mesh [3] or a single distribution tree [4]. Resilience to node failures and error prone transmission paths appeared as important criteria later.

Robustness to node churns, i.e. node departures that disturb the data flow, was considered in SRMS [5] by distributing packets to randomly chosen neighbors outside of the distribution tree. Though this scheme provides some resilience to losses, it is known that repeating information is less efficient than using error correcting codes. SplitStream [6] and CoopNet [1] introduce multiple distribution trees and employ priority encoding transmission (PET) [7] based on forward error correction (FEC) [8] to decrease the effects of node failures and to recover from packet losses. Simulations were used to show the resilience of these schemes under various scenarios showing that increasing the number of trees improves the resilience both to packet losses and node churns.

Feasibility issues of small overlays with less than 100 nodes were discussed in detail in [9] based on experimental broadcasts over the Internet, and showed promising results. The experiments showed that poor performance was due in a large extent to packet losses. The feasibility of larger deployments was studied for a CoopNet like end-node overlay via simulations based on measured traces of user behavior in [2]. The authors concluded that application layer multicast architectures have enough resources, are stable in spite of group dynamics and hence can support large scale streaming content distribution.

Albeit there is an extensive literature on end-point-based multicast streaming, previous work on the behavior of these systems was limited to simulations. In this paper we present a simple model for a CoopNet like overlay combined with FEC, and evaluate the performance of such a system for a large number of nodes. We consider correlated and inhomogeneous losses and the effects of group dynamics. Our results show that an arbitrarily high packet reception probability can be achieved independent of the number of nodes in the overlay by adding enough redundancy (while keeping the bitrate constant). The packet reception probability goes however to zero if there is not enough redundancy added. The transition between the stable and non-stable states of the system is ungraceful, which can raise problems in a dynamic environment.

The paper is organized as follows. In Section 2 we briefly describe the architecture of the considered end-point-based application overlay for multicast. In Section 3 we present the mathematical model and the main results. In Section 4 we discuss the performance of the system based on the analytical model and simulations. In Section 5 we conclude our work.

2 System Description

We consider an application overlay as the one described in [1, 2] consisting of a root node and N peer nodes. Peer nodes are organized in t distribution trees, either by a distributed protocol or a central entity like in [1]. The nodes are members of all t trees, and in each tree they have a different parent node from which they receive data and a different child node to which they forward data. Child nodes of the root node can have the same parent (i.e. the root) in more than one tree. Upon construction of the distribution trees each node is at the same distance from the root node in all trees, and we will refer to nodes at distance i nodes from the root as members of layer i. In the presence of group dynamics it is the task of the tree building algorithm to ensure that all parent nodes of a node are in the same or almost the same layer. We denote the number of children of the root node in each tree by m, and we call it the multiplicity of the root node. The number of layers in the distribution tree is N/m. Typically the number of distribution trees is no more than the multiplicity of the root node $m \geq t$; we will consider this case in the analysis. We assume that nodes do not contribute more bandwidth towards their children as they use to download from their parents, so that the multiplicity of the peer nodes is one, i.e. they have one child in each distribution tree (See Fig. 1).

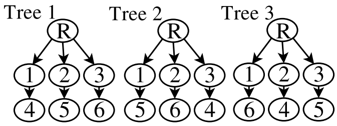


Fig. 1. Multicast tree structure for $t = 3$, $m = 3$ and $N = 6$

The root uses block based FEC, e.g. Reed-Solomon codes, so that nodes can recover from packet losses due to network congestion and node departures. To every k packets of information c packets of redundant information are added resulting in a block length of $n = k + c$. If a source would like to increase the ratio of redundancy while maintaining its bitrate unchanged, then it has to decrease its source rate. We denote this FEC scheme by FEC(n,k). Using this FEC scheme one can implement UXP, PET or the MDC scheme considered in [1]. Lost packets can be reconstructed as long as no more than c packets are lost out of n packets. The root sends every t^{th} packet to its children in a given tree. If $n \leq t$ then at most one packet of a block is distributed over the same distribution tree. Peer nodes relay the packets upon reception to their respective child nodes in the tree corresponding to the particular packets, and once they received at least k packets of a block of n packets they recover the remaining c packets and send them to the child nodes in the corresponding distribution trees. A packet received from the parent node after it has been decoded based on other packets in the block will be discarded.

3 Mathematical Model

In this section we present a mathematical model that describes the behavior of the system in the presence of packet losses due to congestion in the network. Our goal is to calculate the probability $\pi(i)$ that a node in layer i of the distribution tree receives or can reconstruct an arbitrary packet, where i can be arbitrarily high. We model the correlated losses at the input-link of the nodes by a two-state time-discrete Markovian model, often referred to as the Gilbert model [10]. We denote the probability that a packet is lost on the path between two adjacent peer nodes by p_ω $(0 < p_\omega < 1)$. The probability that a packet is lost on the input link of a node given that the previous packet from the same block of packets was lost is denoted by $p_{\omega|\omega}$. The parameters p and q of the Gilbert model are calculated as $q = 1 - p_{\omega|\omega}$ and $p = \frac{p_\omega q}{1-p_\omega}$. Based on the Gilbert model we can calculate the probability of loosing l packets out of j consecutive packets denoted by $P(l, j)$ [10]. Losses seen by different nodes are assumed to be independent. We assume that the probability that a node is in possession of a packet is independent of the probability that another node in the same layer possesses a packet from the same block of packets. We will comment on the validity and possible effects of these assumptions later.

In the following we give a nonlinear recurrence equation [11] to calculate the evolution of $\pi(i)$. As the root node possesses all packets, the initial condition is

$$\pi(0) = 1. \tag{1}$$

Consider the n packets of an FEC block that should arrive from different parents to a node in layer $i+1$. The average number of packets received or reconstructed at the node can be calculated as the average number of packets reconstructed given that j packets have been transmitted from the parents multiplied by the probability that the parents possess j out of the n packets. The probability that a node in layer $i+1$ $(i \geq 0)$ will possess a packet can then be calculated as

$$\pi(i+1) = R(\pi(i), p_\omega, p_{\omega|\omega}) = \sum_{j=1}^{n} \binom{n}{j} \pi(i)^j (1-\pi(i))^{n-j} \frac{1}{n} \sum_{l=1}^{j} \tau(l) P(j-l, j), \tag{2}$$

where $\tau(l)$ indicates the number of packets after FEC reconstruction if l packets have been received and is given as

$$\tau(l) = \begin{cases} l & 0 \leq l < k \\ n & k \leq l \leq n. \end{cases}$$

If losses occur independently on the input links of the nodes then $P(l, j) = \binom{j}{l} p_\omega^l (1-p_\omega)^{j-l}$, and the model becomes the same as the one presented in [12] for independent losses.

We can rewrite (2) by subtracting $\pi(i)$ from both sides and omitting the indices to

$$f(\pi) = -\pi + \frac{1}{n} \sum_{j=1}^{n} \binom{n}{j} \pi^j (1-\pi)^{n-j} \sum_{l=1}^{j} \tau(l) P(j-l, j). \tag{3}$$

Figs. 2 and 3 show examples of $f(\pi)$ for independent and correlated losses respectively. Since $\pi(i+1) - \pi(i) = f(\pi(i))$ we have that for any layer i if $f(\pi(i)) < 0$ then $\pi(i+1) < \pi(i)$, if $f(\pi(i)) > 0$ then $\pi(i+1) > \pi(i)$ and if $f(\pi(i)) = 0$ then

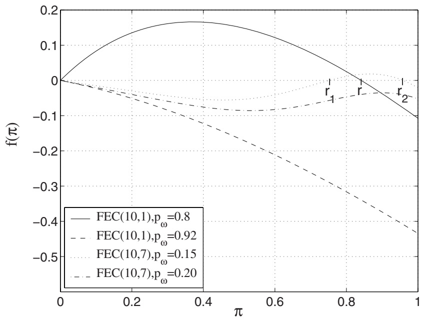


Fig. 2. $f(\pi)$ vs. π for different ratios of redundancy and loss probabilities, independent losses (p+q=1). At the root in (0,1) closest to 1 (if it exists) the derivative is negative and hence the corresponding fixed point is stable.

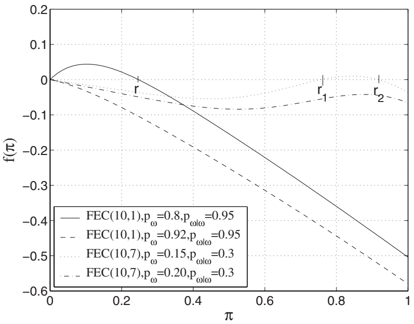


Fig. 3. $f(\pi)$ vs. π for different ratios of redundancy and loss probabilities, correlated losses. At the root in (0,1) closest to 1 (if it exists) the derivative is negative and hence the corresponding fixed point is stable.

$\pi(i+1) = \pi(i)$ and $\pi(i)$ is a fixed point of (2). Starting with $\pi(0) = 1$ as in eq. (1) the value of $\pi(i)$ will decrease as long as $f(\pi(i)) < 0$. The roots of $f(\pi)$ correspond to the fixed points of eq. (2). If $f(\pi)$ has a real root r in the interval $(0,1)$ and the derivative $f^{(1)}(r) = \frac{d}{d\pi} f(\pi)\big|_{\pi=r} < 0$ then $\pi(\infty) = \lim_{i\to\infty} \pi(i) = r$ (e.g. r and r_2 in Figs. 2 and 3), since the fixed point corresponding to this root is asymptotically stable. A fixed point corresponding to a root r with $f^{(1)}(r) > 0$ is unstable on the other hand (e.g. r_1 in Figs. 2 and 3). If $f(\pi)$ does not have real a root in $(0,1)$ then $\pi(\infty) = 0$, since $f(\pi)$ is always negative on $(0,1)$ as we show it later (e.g. the dashed line in Figs. 2 and 3). We will call the system stable if $\pi(\infty) > 0$ and unstable otherwise. To check the existence and the number of real roots of $f(\pi)$ in $(0,1)$ we investigate the signs of $f(\pi)$ at the endpoints of the interval.

For any $p_\omega > 0$ the ratio of successfully received or recovered packets has to be less than 1, so that $f(1) < 0$. Since $\pi = 0$ is a zero of $f(\pi)$ we have to calculate $f^{(1)}(0)$ to see the sign of $f(0+) = \lim_{\pi\to 0+} f(\pi)$.

If $k = 1$ then we get that $f^{(1)}(0) = n(1 - P(1,1)) - k$, and thus if $P(1,1) = p_\omega < (n-k)/n$ then $f^{(1)}(0) > 0$ and consequently $f(0+) > 0$. Hence there has to be at least one root r in $(0,1)$ for which $f^{(1)}(r) < 0$ resulting in an asymptotically stable fixed point (e.g. the solid line in Figs. 2 and 3). It follows that for any loss probability p_ω there is a ratio of redundancy c/k above which $\lim_{i\to\infty} \pi(i) > 0$. Otherwise, if $p_\omega \geq (n-k)/n$, then the number of real roots in $(0,1)$ is either zero or an even number, and using Sturm's theorem [13] we find that the number of roots in $(0,1)$ is 0 for any such p_ω (e.g. the dashed line in Figs. 2 and 3).

If $k > 1$ then we have $f^{(1)}(0) = -P(1,1) = -p_\omega$, which is always negative, and thus the number of real roots in $(0,1)$ is either zero or an even number. By using Sturm's theorem we find for any p_ω and $p_{\omega|\omega}$ that the number of real roots in $(0, 1)$ is no more than two (counting their multiplicity). If they exist, denoted by r_1 and r_2 ($r_1 \leq r_2$), then $f^{(1)}(r_2) < 0$, r_2 is asymptotically stable and $\pi(\infty) = r_2$ (e.g. the dotted line in Figs. 2 and 3). Since $f(\pi) > 0$ for $r_1 < \pi < r_2$, the above result holds for any $r_1 < \pi(0) \leq 1$ as initial condition. Similarly, even if $r_1 < \pi(i) < r_2$ for some i, we have $\pi(\infty) = r_2$. With other words, the system can recover from disturbances, as long as $\pi(i) > r_1$. Let us denote the bifurcation point in p_ω at which the asymptotically stable fixed point (r_2) annihilates with the unstable fixed point (r_1) and both disappear by $p_{max}(p_{\omega|\omega})$. For $0 < p_\omega < p_{max}(p_{\omega|\omega})$ we have that $0 < r_1 < r_2 < 1$. However, $f(\pi)$ has no roots in $(0, 1)$ for $p_\omega > p_{max}(p_{\omega|\omega})$ (e.g. the dash-dotted line in Figs. 2 and 3). In the special case when losses are uncorrelated, i.e. $p_{\omega|\omega} = p_\omega$, we will denote the bifurcation point in p_ω by p_{max}.

In the following we discuss the validity and effects of certain assumptions made in the model. The model considers loss correlations at the input links of the nodes. If losses occur in bursts at the output links of the nodes, the burstiness influences the results if packets from the same block are distributed over the same distribution tree, i.e. $t < n$, but does not influence them otherwise. We do not consider this case in this analysis. The assumptions $n \leq t$ and $m \geq t$ are made to ensure independence of the losses of packets in the same block and to ensure that each node has different parents in all of the trees respectively. Removing these assumptions will make losses more correlated, and hence worsen the performance of the distribution tree. On the other hand, setting $t > n$ will not improve the performance of the system compared to $t = n$. Hence, when choosing n and t there are two factors to be considered: the delay introduced by an FEC block of length n and the administrative overhead of maintaining t distribution trees.

4 Performance Evaluation

In this section we show results obtained with the model presented in the previous section and simulations. In all scenarios we set $t = n$ and we consider $m = 80$ to $m = 320$ for easy comparison. For the simulations we considered the streaming of a 112.8 kbps stream to nodes organized in 1000 layers, hence the number of nodes in the overlay is between 80000 and 320000. The packet size is 1410 bytes. The peer nodes have 128 kbps connections both uplink and downlink. Nodes choose their parent nodes at random, and avoid having the same parents in different trees whenever possible. During each run of the simulation the root node sends about $10000 \times m$ to $30000 \times m$ packets, so that there are 0.8 to 9.6 million packets sent per layer, and 0.8 to 9.6 billion packets sent in the overlay.

4.1 Independent and Homogeneous Losses

We start the evaluation by considering the simplest scenario, homogeneous, uncorrelated losses. All nodes experience the same packet loss probability, and packets arriving to a particular node are lost independent from each other.

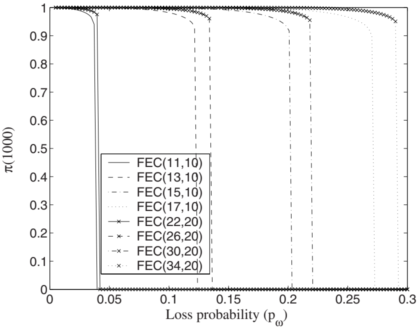


Fig. 4. $\pi(1000)$ vs p_ω for different ratios of redundancy

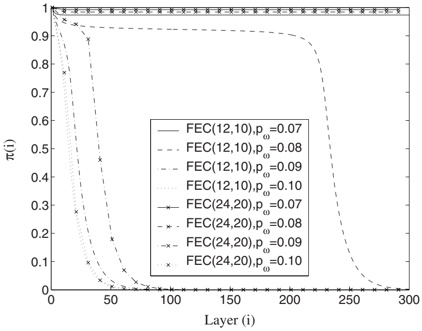


Fig. 5. $\pi(i)$ vs. i for different ratios of redundancy and loss probabilities

Figure 4 shows $\pi(1000)$ as a function of p_ω for $k = 10$ and $k = 20$ and different values of c. The figure shows that for every (n, k) pair there is a loss probability p_{max} above which the reception probability in nodes far from the root node suddenly becomes 0. Below p_{max} the reception probability is close to 1 and is slowly decreasing. This stepwise, ungraceful decrease of the reception probability is an undesired feature for systems working in a dynamic environment such as the Internet. The figure shows that increasing the number of trees, i.e. the FEC block length, slightly improves the resilience of the distribution tree to losses, which is in accordance with [1, 8].

Figure 5 shows $\pi(i)$ as a function of i for different block lengths n and loss probabilities and a ratio of redundancy of $c/k = 0.2$. We see that $\pi(i)$ is close to one in the cases when $p_\omega < p_{max}$, while it becomes almost 0 after some i otherwise. The value of i at which $\pi(i)$ breaks down depends on how far p_ω is from p_{max}. For $k = 10$ $p_{max} = 0.0799$ and for $k = 20$ $p_{max} = 0.0885$. The positive effects of the increased block length can be seen by comparing results

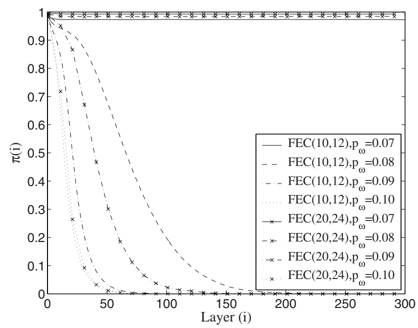


Fig. 6. $\pi(i)$ vs. i for different ratios of redundancy and loss probabilities, simulation results

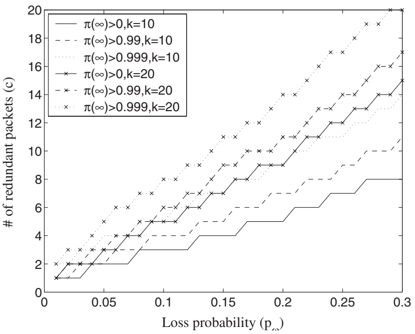


Fig. 7. Number of redundant packets vs. p_ω for different performance objectives

at $p_\omega = 0.08$, where for $k = 20$ the system is stable, whereas for $k = 10$ it is unstable. Figure 6 shows simulation results for the same scenarios as Fig. 5. The comparison shows perfect match for $p_\omega = 0.07$ and $p_\omega = 0.10$, while the simulation results show worse behavior than the analytical ones when p_ω is close to p_{max}. The difference is rather big for $k = 10$ and $p_\omega = 0.08$, when $p_\omega - p_{max} = 10^{-4}$, in which case deviations from the mean loss probability in the individual layers make the deterioration faster than that predicted by the model. For higher values of m the simulation gives more accurate results as the probability of deviation is lower due to the central limit theorem.

Figure 7 shows c, the number of redundant packets needed to ensure $\pi(\infty) > 0$, $\pi(\infty) > 0.99$ and $\pi(\infty) > 0.999$ for $k = 10$ and $k = 20$. The figure shows a closely linear relationship between the number of redundant packets needed and the loss probability. For low values of p_ω the number of redundant packets needed to ensure $\pi(\infty) > 0.999$ is close to the number of redundant packets needed for $\pi(\infty) > 0$, and hence in a dynamic environment the ratio of redundancy has to be set higher to prevent a severe decrease of $\pi(i)$ due to a sudden increase of the loss probability.

4.2 Inhomogeneous Losses

In this subsection we consider the scenario when the packet loss probability experienced by the individual nodes is not homogeneous, i.e. different nodes experience different loss probabilities. We denote by Q the joint distribution function of p_ω and $p_{\omega|\omega}$ experienced by individual nodes. If the multiplicity m of the root node is high, then the evolution of the packet reception probability can be described by the equation

$$\pi(i+1) = \int_Q R(\pi(i), p_\omega, p_{\omega|\omega}) dQ, \tag{4}$$

where $R(\pi(i), p_\omega, p_{\omega|\omega})$ was defined in eq. 2. This approximate model treats layer i as homogeneous, and calculates the mean of the packet reception probability in layer $i+1$ given the distribution Q of the packet loss probability. In the following we consider non-correlated losses (p+q=1) to keep the number of parameters low and we show results for two cases of inhomogeneous losses. In the first, bimodal case, γ_l portion of the nodes experiences loss probability p_ω^l and the rest p_ω^h, the mean packet loss probability in the overlay is $p_\omega = \gamma_l p_\omega^l + (1-\gamma_l) p_\omega^h$. In the second, uniform case, the loss probabilities are uniformly distributed between p_ω^l and p_ω^h, the mean packet loss probability in the overlay is $p_\omega = (p_\omega^l + p_\omega^h)/2$. For easy comparison we consider $n = 12$, FEC(12,10) and $m = 160$. We do not consider $p_\omega^h > 0.1$, as nodes that experience higher loss probabilities will leave the overlay due to poor quality. Figure 8 shows results obtained with the model for various bimodal and uniform distributions. The figure shows that the presence of nodes with high loss probabilities decreases $\pi(i)$ compared to the homogeneous case. This is due to that R is a concave function of p_ω and π for values of interest of p_ω ($p_\omega < 0.1$). Simulation results shown in Figure 9 for

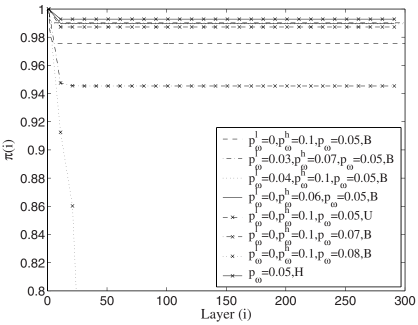


Fig. 8. $\pi(i)$ vs i for inhomogeneous losses

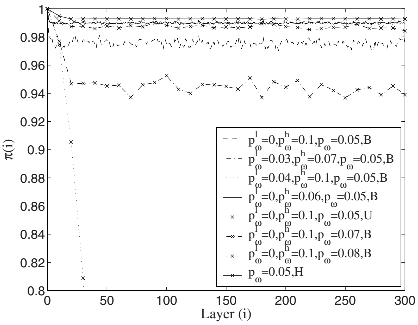


Fig. 9. $\pi(i)$ vs i for inhomogeneous losses. Simulation results.

$\pi(i)$ as a function of i show a good match with the mathematical model. Since the number of nodes in each layer is finite, the mean packet loss probability in individual layers can deviate from the mean loss probability in the overlay, which explains the high variance of the simulation results. The variance of the results decreases as m increases due to the central limit theorem.

4.3 Correlated Losses

Figure 10 shows $\pi(1000)$ as a function of p_ω for various values of $p_{\omega|\omega}$ obtained with the model. The figure shows that the value of both r_2 and $p_{max}(p_{\omega|\omega})$ decreases as losses become more correlated (i.e. $p_{\omega|\omega}$ increases). Figure 11 shows $\pi(i)$ as a function of p_ω for correlated losses for scenarios similar to those in Fig. 5 as obtained with the model. Comparing the two figures shows that correlated losses decrease the packet reception probability significantly whenever the system is stable. Figure 12 shows matching simulation results for the same scenarios.

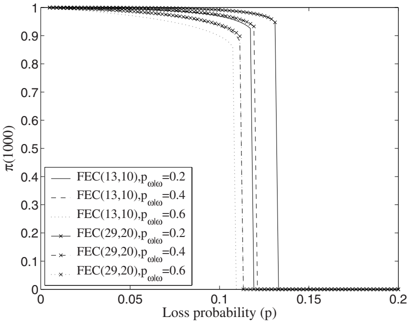


Fig. 10. $\pi(1000)$ vs. p_ω for m=80, FEC (13,10) and FEC(26,20) and different values of $p_{\omega|\omega}$

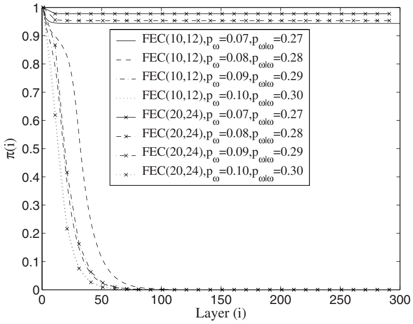


Fig. 11. $\pi(i)$ vs i for different ratios of redundancy and loss probabilities, correlated losses

4.4 Malicious Layers

In this subsection we investigate how the presence of layers with extreme loss probabilities (e.g. a DDoS attack) influences the packet reception probability. We consider an overlay, where the loss probability is p_ω, except for layers 50 and 100, which experience significantly higher (p_ω^m) packet loss probability. Such layers decrease $\pi(i)$ below $r_2(p_\omega)$ (the stable fixed point corresponding to p_ω) in the layers following them. Based on Figs. 2 and 3 we expect that the overlay can recover as long as $\pi(i)$ remains larger than $r_1(p_\omega)$. Figure 13 shows $\pi(i)$ as a function of i for $p_\omega = 0.05$. The figure shows that for $m = 80$ the malicious layers influence the packet reception probability for all following layers already at $p_\omega^m = 0.25$, while for $m = 320$ the overlay is able to recover. The different behavior is due to that if less than $r_1(0.05) = 0.7499$ portion of the packets of an FEC block is received in the malicious layers, then that block will be entirely lost in the following layers. For $m = 320$ the probability that less than $r_1(0.05)$ portion of the packets in a block will be received in layers 50 or 100 is only significant for $p_\omega^m = 0.4$, while for $m = 80$ this probability is significant already for $p_\omega^m = 0.25$ (around 0.05). Hence, once again, increasing m improves the robustness of the overlay.

4.5 Effects of Group Dynamics

In this subsection we analyze the effects of node departures on the packet reception probability. We assume that the departure of a node interrupts the flow of data to its child nodes for a random time T. This time T includes the time it takes for the child node to notice that its parent node has departed and the time it takes to find a new parent node. For a description of how the departure of a parent or a child node can be detected see [1]. Several algorithms have been proposed to find a suitable parent node, a comparison of some simulation results is shown in [2].

In this work we consider an ideal parent selection algorithm that maintains the structure of the overlay despite of the node departures. Arriving nodes take the places of the departed nodes, and hence fill the gaps in the distribution tree. We consider the stationary state of the system, when the arrival and departure rates are equal. We assume that the interarrival times are exponentially distributed, this assumption is supported by several measurement studies [14, 15]. The distribution of the session holding times has been shown to fit the log-normal distribution [14].

Based on the model presented in Section 3 we expect that node departures can be included in the model as an increase of the packet loss probability as $p_\omega^d = N_d/N \times \overline{T}$, where N_d is the mean number of nodes departing per time unit and $\overline{T}$ is the mean of the time nodes need to recover from the departure of a parent node. The rationale for this hypothesis is that node departures can be treated as bursty losses on the output link of the departing nodes, and can be modeled as independent if $n \leq t$ and $m \geq t$.

To simulate the ideal tree construction algorithm, instead of removing the departing and inserting the arriving nodes, we switch nodes off after their session

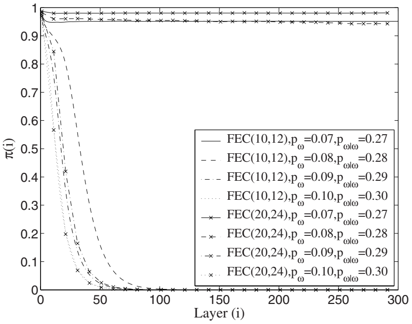


Fig. 12. $\pi(i)$ vs i for different ratios of redundancy and loss probabilities, correlated losses. Simulation results.

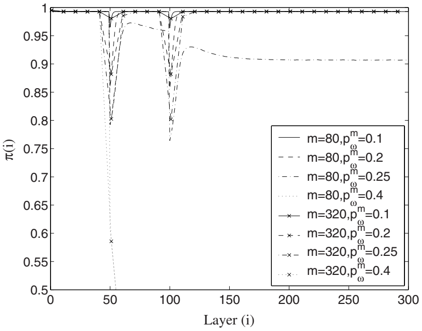


Fig. 13. $\pi(i)$ vs i in the case of malicious layers. Simulation results.

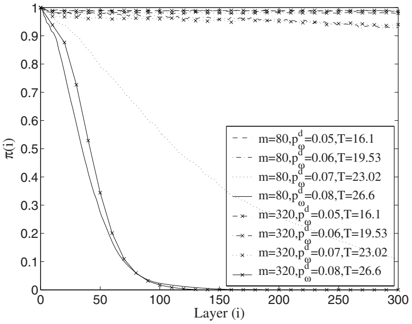


Fig. 14. $\pi(i)$ vs i for $1/\mu = 306s$. Simulation results.

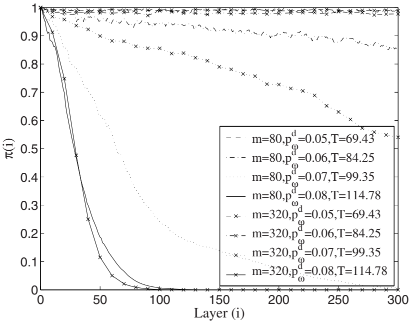


Fig. 15. $\pi(i)$ vs i for $1/\mu = 1320s$. Simulation results.

holding time has elapsed, and switch them on after a random time T, which would correspond to the reconstruction of the tree. We can change the value of p_ω^d by adjusting T and the session holding time $1/\mu$. We show simulation results for $m = 80$ and $m = 320$, and we consider two mean session holding times, $1/\mu = 306s$ as measured in [14] and $1/\mu = 1320s$ as measured in [2]. The parameters of the corresponding log-normal distributions are $M = 4.93, S = 1.26$ and $M = 5.46, S = 1.85$ respectively.

Figs. 14 and 15 show results for $1/\mu = 306s$ and $1/\mu = 1320s$ respectively considering FEC(12,10). The two figures show the same characteristics despite of the different mean session holding times, which supports our approximation. The figures show that for $p_\omega^d = 0.05$ the overlay is stable for both $m = 80$ and $m = 320$, for $p_\omega^d = 0.06$ the overlay is only stable for $m = 320$, while for higher values of p_ω^d it is unstable. The overlay would become stable for $p_\omega^d = 0.07$ by increasing the value of m similar to the results shown in Fig. 5 obtained with the analytical model. To understand why increasing the number of nodes per layer (m) gives better resilience to node departures, let us consider the evolution of the number of active nodes per layer (ν). If ν/m in a layer is lower than $1 - p_{max}$ then it is likely that the following layers will not be able to recover the missing data. The evolution of ν can be modeled by an Engset system [16] (due to its insensitivity to the distribution of the service time, i.e. the nodes' lifetime distribution). ν follows a binomial distribution with parameters m and $\beta = \mu/(1 + \mu T)$, its mean is βm and its coefficient of variation (the ratio of the standard deviation and the mean) is $\sqrt{(1 - \beta)/(m\beta)}$. Consequently, the higher the number of nodes per layer, the lower the probability that the ratio of active nodes is lower than $1 - p_{max}$, which explains the improved performance of the overlay as m increases.

Node departures can have an aggravating effect towards the end of a broadcast, when departures cause the number of nodes in the overlay to decrease. The increased loss probability due to node churn might exceed the level of stable operation and can lead to $\pi(\infty) = 0$. The root node can prevent this from happening by increasing the ratio of redundancy c/k towards the end of the broadcast.

5 Conclusion

In this paper we presented a mathematical model for the analysis of an end-point overlay for multicast based on multiple distribution trees and forward error correction. We showed that for any loss probability there is a ratio of redundancy which ensures that even nodes far away from the root of the trees receive a non-zero ratio of the information. We showed that this multicast scheme shows a non-graceful performance degradation once the loss probability exceeds a certain threshold. The threshold depends on the number of distribution trees and the ratio of redundancy used. Using the model and simulations we showed that correlated and inhomogeneous losses decrease both the ratio of received packets and the value of the stability threshold of the system. We analyzed how malicious layers can influence the behavior of the system, and concluded that increasing the number of nodes per layer gives improved robustness. The performance evaluation in the presence of dynamic node behavior led to the same conclusion. The results presented here show that the ratio of redundancy has to be adjusted with care to maintain the stability of this overlay, as underestimating the loss probability in the network can lead to the loss of all data.

References

1. V. Padmanabhan, H. Wang, and P. Chou, "Resilient peer-to-peer streaming," in *Proc. of IEEE ICNP*, pp. 16–27, 2003.
2. K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, "The feasibility of supporting large-scale live streaming applications with dynamic application end-points," in *Proc. of ACM SIGCOMM*, pp. 107–120, 2004.
3. Y. Chu, S. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE J. Select. Areas Commun.*, vol. 20, no. 8, 2002.

4. S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proc. of ACM SIGCOMM*, 2002.
5. S. Banerjee, S. Lee, R. Braud, B. Bhattacharjee, and A. Srinivasan, "Scalable resilient media streaming," in *Proc. of NOSSDAV*, 2004.
6. M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High bandwidth content distribution in a cooperative environment," in *Proc. of IPTPS*, 2003.
7. B. Lamparter, A. Albanese, M. Kalfane, and M. Luby, "PET - priority encoding transmission: A new, robust and efficient video broadcast technology," in *Proc. of ACM Multimedia*, 1995.
8. K. Kawahara, K. Kumazoe, T. Takine, and Y. Oie, "Forward error correction in ATM networks: An analysis of cell loss distribution in a block," in *Proc. of IEEE INFOCOM*, pp. 1150–1159, June 1994.
9. Y. Chu, A. Ganjam, T. Ng, S. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early experience with an Internet broadcast system based on overlay multicast," in *Proc. of USENIX*, 2004.
10. E. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, September 1963.
11. I. Gumowski and C. Mira, *Recurrences and Discrete Dynamic Systems, LNM-809.* Springer-Verlag, 1980.
12. G. Dán, V. Fodor, and G. Karlsson, "On the asymptotic behavior of end-point-based multimedia streaming," in *Proc. of International Zürich Seminar on Communication*, 2006.
13. S. Basu, R. Pollack, and M. Roy, *Algorithms in real algebraic geometry.* Springer Verlag, 2003.
14. E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," in *Proc. of ACM IMC*, pp. 117–130, 2002.
15. K. Sripanidkulchai, B. Maggs, and H. Zhang, "An analysis of live streaming workloads on the Internet," in *Proc. of ACM IMC*, pp. 41–54, 2004.
16. L. Kleinrock, *Queueing Systems*, vol. I. Wiley, New York, 1975.

Multicast Tree Aggregation in Large Domains

Joanna Moulierac[1], Alexandre Guitton[2], and Miklós Molnár[3]

[1] IRISA/University of Rennes I, 35042 Rennes, France
[2] Birkbeck College, University of London, England
[3] IRISA/INSA, Rennes, France
joanna.moulierac@irisa.fr,
alexandre@dcs.bbk.ac.uk, miklos.molnar@irisa.fr

Abstract. Tree aggregation is an efficient proposition that can solve the problem of multicast forwarding state scalability. The main idea of tree aggregation is to force several groups to share the same delivery tree: in this way, the number of multicast forwarding states per router is reduced. Unfortunately, when achieving tree aggregation in large domains, few groups share the same tree and the aggregation ratio is small. In this paper, we propose a new algorithm called TALD (Tree Aggregation in Large Domains) that achieves tree aggregation in domains with a large number of nodes. The principle of TALD is to divide the domain into several sub-domains and to achieve the aggregation in each of the sub-domain separately. In this way, there is possible aggregation in each of the sub-domain and the number of forwarding states is significantly reduced. We show the performance of our algorithm by simulations on a Rocketfuel network of 200 routers.

Keywords: Multicasting, tree aggregation, network simulation.

1 Introduction

With the growth of the number of network applications, it has been found a few years ago that the bandwidth was a bottleneck. Multicast has been developed to spare the bandwidth by sending efficiently copies of a message to several destinations. Although many research has been done on multicast, its deployment on the Internet is still an issue. This is due mainly to the large number of multicast forwarding states and to the control explosion when there are several concurrent multicast groups. Indeed, in the current multicast model, the number of multicast forwarding states is proportional to the number of multicast groups. The number of multicast groups is expected to grow tremendously together with the number of forwarding states: this will slow down the routing and saturate the routers memory. Additionally, the number of control messages required to maintain the forwarding states will grow in the same manner. This scalability issue has to be solved before multicast can be deployed over the Internet.

Tree aggregation is a recent proposition that greatly reduces both the number of multicast forwarding states and the number of control messages required to maintain them. To achieve this reduction, tree aggregation forces several groups

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 691–702, 2006.

to share the same multicast tree. In this way, the number of multicast forwarding states depends on the number of trees and not on the number of groups.

1.1 Tree Aggregation

The performance of tree aggregation mechanisms depends on how different groups are aggregated to the same tree within a domain. To aggregate several groups to the same tree, a label corresponding to the tree is assigned to all the multicast packets at the ingress routers of the domain. In the domain, the packets are forwarded according to this label. The label is removed at the egress routers so that the packets can be forwarded outside the domain. In addition to the multicast forwarding states that allow to match an incoming label to a set of outgoing interfaces, the border routers of the domain have to store group-specific entries. A group-specific entry matches a multicast address with a label.

Let us show the tree aggregation mechanism on an example. Figure 1 represents a domain with four border routers and the group-label table of the border router b_1. The two groups g_1 and g_3 can be aggregated to the same tree corresponding to label l_1 while g_2 uses its own label l_2. If a new group g_4 has members attached to routers b_1 and b_4, the tree manager can also aggregate g_4 to label l_1 or to the label l_2. In this case, no new tree is built but bandwidth is wasted with l_1 (resp. with l_2) when the messages for g_4 reach b_2 (resp. reach b_3) unnecessarily. Otherwise, the tree manager can build a new tree with label l_3 for g_4. In this case, no bandwidth is wasted but more forwarding states are required. Therefore, there is a trade-off between the wasted bandwidth and the number of states.

1.2 Limits of Tree Aggregation in Large Domains

With tree aggregation, the number of forwarding states is proportional to the number of trees and not to the number of groups as in traditional multicast.

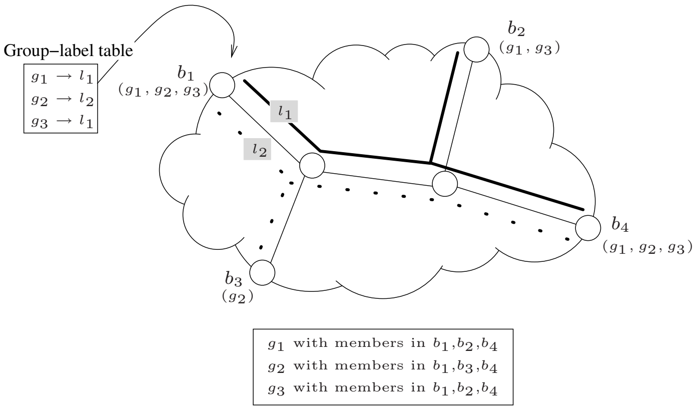


Fig. 1. Tree aggregation in a small domain

However, when the domain is too large, we show that tree aggregation builds as many trees as traditional multicast and that there is no reduction of the number of forwarding states. Indeed, the number of different groups increase with the number b of border routers in the domain and the number g of concurrent groups. Therefore, it can be identified as the expected number of non-empty urns obtained by randomly throwing g balls into 2^b urns :

$$\textit{Number of different groups} = 2^b(1 - (1 - 2^{-b})^g)$$

This formula gives the total number of different groups in a domain with b border routers when there are g concurrent groups. We assume that the size of each group is chosen uniformly and that the members of each groups are chosen uniformly. This assumptions correspond to the worst-case scenario, where there is no correlation between groups.

Consequently, when there are too many border routers, the number of different groups is too large and the probability of finding a tree already existing for a new group is low. We will show in this paper that the existing protocols achieve tree aggregation within small domains of around 20 to 40 border routers but perform few aggregations in larger domain. Consequently, in large domains, the number of forwarding states is not reduced compared to traditional multicast and a new protocol has to be proposed in order to manager the large domains.

1.3 Proposition : Sub-domain Tree Aggregation

In this paper, we propose a new protocol that performs aggregations in large domains. This protocol, TALD, for Tree Aggregation in Large Domains, divides the network into several sub-domains before aggregating. In this way, aggregation is feasible in each of the sub-domain and the number of forwarding states is strongly reduced.

Let us suppose that the domain is divided into d domains of approximately the same number of nodes. The union of the d domains is equal to D and the domains are disjoint. Thus, the number of different groups can be seen as :

$$\textit{Number of different groups} = 2^{b/d}(1 - (1 - 2^{-b/d})^g) \times d$$

For example, on a network with 15 border routers, there are 8 618 different groups for 10 000 concurrent groups using the formula above. However, on a network with 40 border routers, there are 10 000 different groups for 10 000 concurrent groups. Consequently, if the members of the groups are distributed uniformly, there are not two group with exactly the same members for 10 000 concurrent groups. Now, if the domain is divided into 4 sub-domains with approximately 10 nodes, the total number of different groups is equal to 4000 (there are approximately 1000 different groups for 10 000 concurrent groups for a domain with 10 border routers). Consequently, when the domain is divided into several sub-domains, the number of different groups decreases significantly.

The rest of the paper is organized as follows. Section 2 describes an algorithm that divides the domain into several sub-domains and describes the aggregation protocol TALD. Section 3 validates the algorithm by showing its performance on simulations. Section 4 describes the existing protocols for tree aggregation. Section 5 concludes and gives the perspectives of our work.

2 The Protocol TALD

In this section, we show how to design the protocol TALD (Tree Aggregation in Large Domains) that achieves sub-domains tree aggregation. Three main issues arise in order to present TALD:

1. How to divide the domains into several sub-domains?
2. How to aggregate groups within a sub-domain?
3. How to route packets in the domain for the multicast group, considering the aggregation of the sub-domains?

2.1 Dividing a Domain into Sub-domains

In order to minimize the total number of different multicast groups, the domain D has to be divided into sub-domains D_i of approximately the same number of nodes. We propose an algorithm that divides the domain $D = (V, E)$ into two sub-domains $D_1 = (V_1, E_1)$ and $D_2 = (V_2, E_2)$ where $V_i \subset V$ is the set of routers of the domain D_i and $E_i \subset E$ the set of links.

The main idea of the algorithm is to find first the two nodes x_1 and x_2 with the maximum distance in the domain D, *i.e.* the two most distant nodes. Then, two sets of nodes V_1 and V_2 are created with $x_1 \in V_1$ and $x_2 \in V_2$. Iteratively, the nearest nodes of the nodes already in the set are added; at each step of the algorithm one node is added in V_1 and one node is added is V_2. When all the nodes of the domain D are whether in V_1 or in V_2, two domains $D_1 = (V_1, E_1)$ and $D_2 = (V_2, E_2)$ are built from the two sets. The edges in E_i are the edges including in E connecting two nodes in V_i. When the two sub-domains have been built, this algorithm can be reapplied on each of the sub-domain in order to get 4 sub-domains or more.

Figure 2 shows the network Eurorings[1] divided into four separated sub-domains by the algorithm presented in this subsection. The network was divided into two sub-domains and then they were also divided into two in order to obtain four separated sub-domains with disjoint sets of nodes of approximately the same size.

2.2 Aggregating in a Sub-domain

We assume in this subsection that the domain is divided into sub-domains. If the domain is already explicitly divided into sub-domains (e.g. for administrative reasons for example), there is no need to apply the algorithm described in the previous subsection.

[1] http://www.cybergeography.org/atlas/kpnqwest_large.jpg

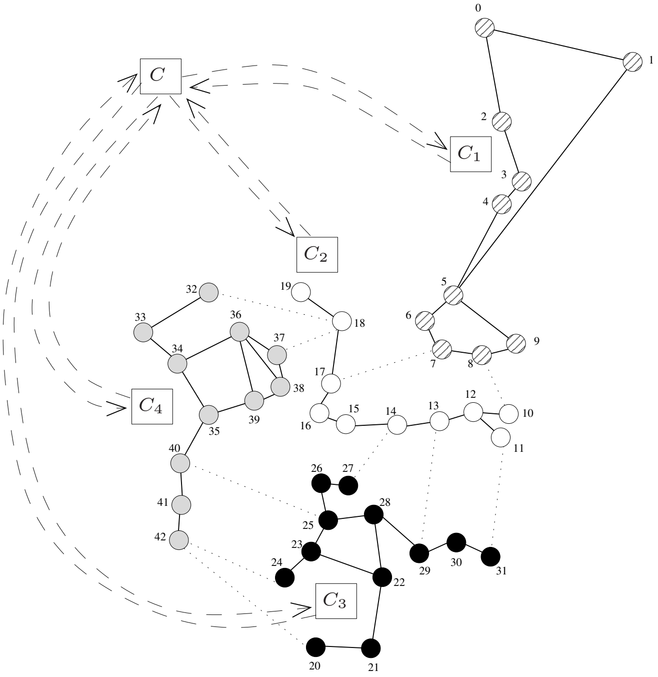


Fig. 2. Eurorings network divided into four sub-domains

Each sub-domain $D_i = (V_i, E_i)$ is controlled by a centralized entity C_i which is in charge of aggregating the groups within the sub-domain. For example, in figure 2, the sub-domain 1 is controlled by C_1. Each C_i knows the topology of the sub-domain (in order to build trees for the multicast groups) and maintains the group memberships for its sub-domain. Note that C_i is aware of only the members in its sub-domains and not the members for all the group.

When a border router receives a `join` or `leave` message for a group g, it forwards it to the centralized entity C_i of its sub-domain in its sub-domain. Then, C_i creates or updates the group specific entries for g in order to route the messages. The centralized entity C_i builds a native tree t_i covering the routers attached to members of g in its sub-domain, and then C_i tries to find an existing tree t_i^{agg} already configured in its sub-domain satisfying these two conditions:

- t_i^{agg} covers all the routers of the sub-domain attached to members of g
- the cost of t_i^{agg} (*i.e.* the sum of the cost of each link of t_i^{agg}) is not more than b_t% of the cost of the native tree t_i where b_t is a given bandwidth threshold:

$$cost(t_i^{agg}) \leq cost(t_i) \times (1 + b_t)$$

The centralized entity C_i chooses among all the trees matching these two conditions the tree t_i^{agg} with minimum cost. Then g is aggregated to t_i^{agg} and C_i updates in all the border routers attached to members of g a group specific entry matching g to t_i^{agg} (or more precisely, matching g to the label corresponding to t_i^{agg}: $g \rightarrow \text{label}(t_i^{agg})$). If no tree satisfies these two conditions, then C_i configures t_i (the tree initially built for g) by adding forwarding states in all the routers covered by t_i and then adds the group specific entries $g \rightarrow \text{label}(t_i)$.

2.3 Routing in the Whole Domain

The centralized entities C_i having members of g in their sub-domains have use the algorithm described in previous subsection. In order to route packets for the whole multicast group, the trees in all the sub-domains have to be connected. The centralized entity C, responsible of the main domain D is in charge of this task. Note that C does not need to know the topology of D to connect these trees. Several solutions are possible to connect these trees. We present in this paper a simple solution to connect these trees in order to validate first the main idea of our algorithm.

In this simple solution, each C_i, having members of g in its sub-domain i, has communicated to C the IP address of one of the routers of the sub-domain attached to members of g. This router is the representative router for g in D_i. The centralized entity C keeps this information and maintains the list of the representatives of g for each sub-domain. Note that C does not keep any information concerning the group memberships. Then, C connect the trees in the sub-domains by adding tunnels. The tunnels can be built by adding group specific entries matching g to routers in the others sub-domains.

For example in figure 2, suppose that C receives a message (g,`@IP(router 5)`) from C_1, a message (g,`@IP(router 11)`) from C_2 and a message (g,`@IP(router 28)`) from C_3. In this example, C has to connect the three trees corresponding to group g in the three sub-domains. In order to achieve this connection, C adds a group specific entry g →`@IP(router 11)` in router 5. Two more are added in router 11: g →`@IP(router 5)` and g →`@IP(router 28)` and one in router 28: g →`@IP(router 11)`. In this way, the three trees in the three sub-domains are connected by tunnels and messages for g can be routed.

As our concerns in this paper is to reduce the number of entries stored, we do not optimize the connection of the trees. This can be done as further part of investigation. What only matters for the moment is the number of group specific entries added. If three C_i have registered members of g to C, four group specific entries are added. More generally, if n C_i have replied to C, then $2(n-1)$ entries are needed.

3 Simulations

We run several simulations on different topologies. Due to lack of space, we present only the results of the simulations on the Rocketfuel graph Exodus [2]. This network contains 201 routers and 434 links. During the simulations, 101 routers were core routers and 100 others routers were border routers and can be

[2] `http://www.cs.washington.edu/research/networking/rocketfuel/`

attached to members of multicast groups. The plots are the results of 100 cases of simulations where each case corresponds to a different set of border routers.

We present the results of the protocols TALD-1, TALD-2 and TALD-4 for different bandwidth thresholds: when 0% bandwidth is allowed to be wasted and when 20% of bandwidth wasted. The protocol TALD-1 represents the actual tree aggregation protocols when the domain is not divided and when aggregation is performed in the main domain. With TALD-2, the domain in divided into 2 sub-domains and with TALD-4, the domain is divided into 4 sub-domains. The division was performed by the algorithm presented in Section 2.1.

The number of multicast concurrent groups varied from 1 to 10 000 and the number of members of groups was randomly chosen between 2 and 20. The members of groups were chosen randomly among the 100 border routers. This behavior is not representative of the reality but it allows to show the performance of the algorithms in worst-case simulation. Indeed, when the members are randomly located, then the aggregation is more difficult than if members of groups are chosen with some affinity model.

3.1 Number of Forwarding States

Figure 3 plots the total number of forwarding states in the domain, *i.e.* the sum of the forwarding states stored by all the routers of the domain. Recall that for a bidirectional tree t, $|t|$ forwarding states have to be stored where $|t|$ denotes the number of routers covered by t. With TALD-1, there is almost no aggregation (the number of forwarding states is the same as if no aggregation was performed) and then, the number of multicast forwarding states is the same with 0% and with 20% of bandwidth wasted. The protocol TALD-4 gives significantly better results than TALD-1 and TALD-2. Moreover, with TALD-4, the number of multicast forwarding states is reduced when the bandwidth threshold is equal to 20%.

For example, TALD-4 stores around 160 000 forwarding states in the whole domain when the bandwidth threshold is equal to 0% for 10 000 concurrent groups. There is a reduction of 22% when the bandwidth threshold is equal to

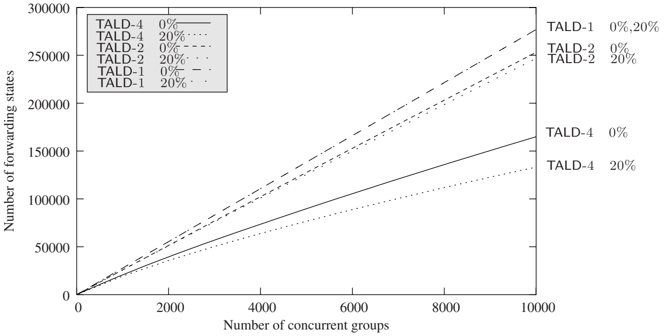


Fig. 3. Number of forwarding states

20%: the number of forwarding states reaches approximately 126 000. Oppositely, the amount of bandwidth wasted has no influence for the results of TALD-1 as the number of forwarding states is the same when 0% of bandwidth is wasted and when 20% of bandwidth is wasted. This shows that traditional aggregation algorithms are not efficient in large domains.

3.2 Group Specific Entries

Figure 4 plots the number of group specific entries which are stored in the group-label table and which match groups to the labels of the aggregated trees. As this number is related to the number of groups, it is not dependent of the bandwidth thresholds and the results are equivalent for 0% and for 20% of bandwidth wasted. The protocols TALD-2 and TALD-4 need to store more group specific entries in order to route the packets for the groups between the sub-domains. These entries are stored in order to configure the tunnels crossing the sub-domains. Consequently, TALD-1 does not store such entries.

The results show that TALD-4 needs to store more entries than TALD-2 which in turn stores more entries than TALD-1. This is the price to be paid to achieve aggregation and to reduce the number of forwarding states. Note that the more sub-domains, the larger the number of groups specific entries. Consequently, it may not be interesting to divide the domain into too many sub-domains because the reduction of forwarding states will not be so significant.

However, TALD-4 reduces the total number of entries stored in routers compared to TALD-1. Figure 5 shows the total number of the groups specific entries and the forwarding states stored in all the routers of the domain. TALD-4 achieves a reduction of 16% of this total number compared to TALD-1 when no bandwidth is wasted and a reduction of 25% with 20% of bandwidth wasted. It may be noted that TALD-2 does not achieved significant reduction of this number compared to TALD-1. Consequently, dividing the domain in two sub-domains is not enough. However, the memory in routers is significantly reduced with TALD-4. As the number of group specific entries increases with the number of sub-domains, it is not be interesting to divide more the domain. Indeed, the more the domain is

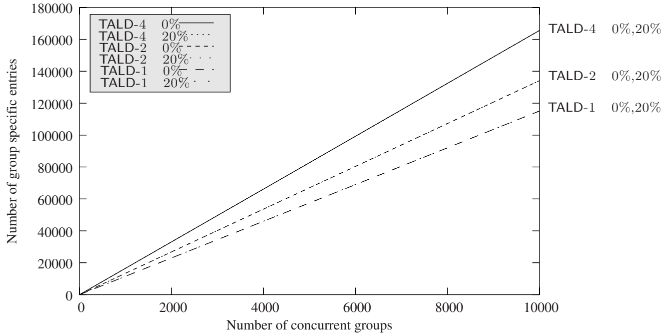


Fig. 4. Group specific entries

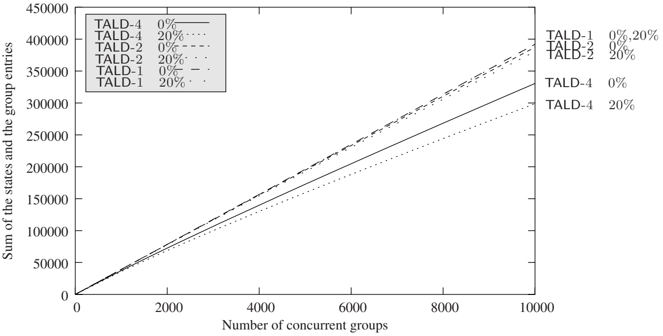


Fig. 5. Sum of the forwarding states and of the group specific entries

divided, the less number of forwarding states but the more the number of group specific entries.

3.3 Aggregation Ratio

Figure 6 shows the aggregation ratio in function of the number of concurrent groups. The aggregation ratio is denoted by the number of trees with aggregation out of the number of trees if no aggregation is performed. Note that for TALD-2 and TALD-4, the number of trees is the sum of the number of trees for each sub-domain.

The protocol TALD-1 achieves less than 1% of aggregation even when 20% of bandwidth is allowed to be wasted. The protocol TALD-4 achieves more than 40% of aggregation even when no bandwidth is allowed to be wasted. When 20% of bandwidth is wasted, the aggregation ratio reaches more than 55%. This figure shows that with large networks, existing algorithms achieving tree aggregation without any division of the domain (as TALD-1) do not realize any aggregation at all.

Figure 7 plots the aggregation ratio in function of the number of border routers in the domain when there are 10 000 concurrent groups. We vary the number of

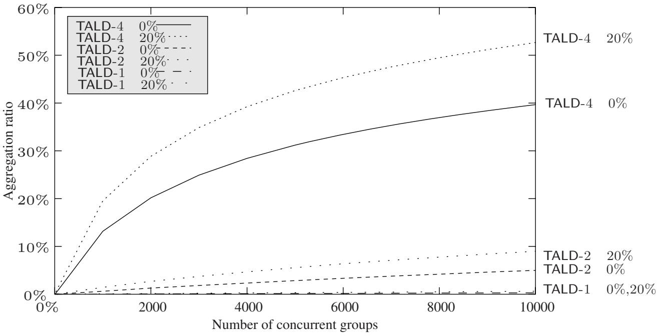


Fig. 6. Aggregation ratio

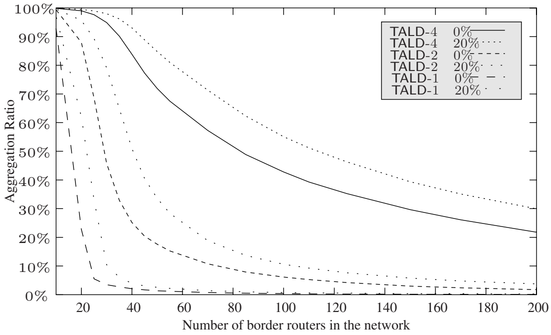


Fig. 7. Aggregation ratio in function of the number of border routers

possible border routers among all the 201 routers of Exodus network from 10 to 200. We run 100 times the algorithm for each possible value of the number of border routers in order to get different sets of border routers. The routers that were not border routers could not be attached to members of multicast groups.

With domains of 10 border routers, the aggregation is very efficient and after 10 000 concurrent groups, the protocols are able to aggregate any new group in the domain. The aggregation ratio decreases dramatically, especially for TALD-1 which is not able to perform any aggregation when the domain contains more than 40 border routers. However, TALD-4 is efficient and performs more than 20% of aggregation even when there are 200 border routers. This shows that for a domain of 40 border routers or more, it is strongly recommended to divide the domain into several sub-domains in order to aggregate groups.

4 Related Work

We presented in this paper, a tree aggregation protocol specific to large domains. In this section, we give an overview of the protocols achieving tree aggregation already in the literature. Tree aggregation idea was first proposed in [5] and since, several propositions have been written.

The protocol AM [2, 3] performs aggregation using a centralized entity called the *tree manager* responsible of assigning labels to groups. The protocol STA [6] proposes to speed up the aggregation algorithm with a fast selection function and an efficient sorting of the trees. These two protocols are represented by TALD-1 during the simulations. TOMA [7] is a recent protocol that performs tree aggregation in overlay networks.

Distributed tree aggregation. The distributed protocol BEAM proposed in [4] configures several routers to take in charge the aggregation in order to ditribute the work load of the *tree manager*. Indeed, in AM or in STA, only the *tree manager* takes this responsibility. The protocol DMTA [9] proposes to distribute the task of the tree manager among the border routers and then to suppress completely the requests to centralized entities necessary in BEAM to achieve aggregation.

Tree aggregation with bandwidth constraints. AQoSM [1] and Q-STA [10] achieve tree aggregation in case of bandwidth constraints. In these two algorithms, links have limited bandwidth capacities and the groups have different bandwidth requirements. Consequently, groups may be refused if no tree can be built satisfying the bandwidth requirements. Q-STA accepts more groups as this protocol builds native tree maximizing the bandwidth available on the links.

Tree aggregation with tree splitting. The protocol AMBTS [8] performs tree splitting before aggregating groups in order to manage larger domains. A tree is divided into several sub-trees and whenever a new group arrives the native tree is splitted in sub-trees according to a foreclosing process. From these sub-trees, the *tree manager* tries to find already existing sub-trees and to aggregate the group. The idea of AMBTS is somehow orthogonal to the idea of TALD. However, we did not compare AMBTS to TALD during the simulations because of the following reasons.

First of all, the protocol is not realistic for large domains as a centralized entity is responsible of all the process of aggregation. This centralized entity keeps the group memberships for all the groups of the whole domain. Moreover, it is in charge of splitting the trees and aggregating the groups. This behavior is not scalable in domains such as Exodus network with 200 routers. Indeed, too much memory is used to store all the information and the centralized entity is strongly solicited each time a member of a group changes. Second, the foreclosing process in which a tree is divided into several sub-trees is not detailed and we were not able to simulate this algorithm due to lack of information. Splitting the trees manually was not possible in our domain. Finally, the number of sub-trees grows tremendously and is larger than the number of groups (especially if the trees are splitted in many sub-trees). Thus, the process of aggregation is strongly slowed down due to the large number of evaluations of sub-trees. In AMBTS, the simulations were done on a network with 16 border routers. All these reasons make us decide to propose and detail a protocol adequate to large domains.

5 Conclusion

In this paper, we proposed a protocol that achieves aggregation in large domains. Indeed, previous known algorithms were not able to perform any aggregation in this case. Consequently, current tree aggregation protocol were not able to reduce the number of forwarding states and behaved in the same way as traditional multicast. The main idea of our protocol is to divide the domain in several sub-domains and to aggregate the groups in each sub-domain. The simulations showed that in large domains where no aggregation was performed, our protocol behaves well and gives good results. The aggregation ratio was around 20% for a domain with 200 border routers while actual protocols achieved 0% of aggregation for domains of more than 40 border routers.

This work leads to many perspectives of research. First, the connection of the trees in each of the sub-domain can be achieved in different ways. Presently, it is done by configuring tunnels however, this connection can be achieved by a tree for example. Second, the domain can be divided using an adaptive algorithm and in more sub-domains, thus it may be interesting to study the impact of this division on the aggregation.

References

1. J.-H. Cui, J. Kim, A. Fei, M. Faloutsos, and M. Gerla. Scalable QoS Multicast Provisioning in Diff-Serv-Supported MPLS Networks. In *IEEE Globecom*, 2002.
2. J.-H. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla. Aggregated multicast, a comparative study. In *IFIP Networking*, number 2497 in LNCS, 2002.
3. J.-H. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla. Aggregated Multicast — A Comparative Study. *Special issue of Cluster Computing: The Journal of Networks, Software and Applications*, 2003.
4. J.-H. Cui, L. Lao, D. Maggiorini, and M. Gerla. BEAM: A Distributed Aggregated Multicast Protocol Using Bi-directional Trees. In *IEEE International Conference on Communications (ICC)*, May 2003.
5. M. Gerla, A. Fei, J.-H. Cui, and M. Faloutsos. Aggregated Multicast for Scalable QoS Multicast Provisioning. In *Tyrrhenian International Workshop on Digital Communications*, September 2001.
6. A. Guitton and J. Moulierac. Scalable Tree Aggregation for Multicast. In *8th International Conference on Telecommunications (ConTEL)*, June 2005.
7. L. Lao, J.-H. Cui, and M. Gerla. TOMA: A Viable Solution for Large-Scale Multicast Service Support. In *IFIP Networking*, number 3462 in LNCS, May 2005.
8. Z.-F. Liu, W.-H. Dou, and Y.-J. Liu. AMBTS: A Scheme of Aggregated Multicast Based on Tree Splitting. In *IFIP Networking*, number 3042 in LNCS, 2004.
9. J. Moulierac and A. Guitton. Distributed Multicast Tree Aggregation. Technical Report 5636, INRIA, July 2005.
10. J. Moulierac and A. Guitton. QoS Scalable Tree Aggregation. In *IFIP Networking*, number 3462 in LNCS, 2005.

Analysis and Performance Evaluation of a Multicast File Transfer Solution for Congested Asymmetric Networks

Pilar Manzanares-Lopez, Juan Carlos Sanchez-Aarnoutse,
Josemaria Malgosa-Sanahuja, and Joan Garcia-Haro

Department of Information Technologies and Communications,
Antiguo Cuartel de Antigones, E-30202, Cartagena, Spain
{pilar.manzanares, juanc.sanchez, josem.malgosa, joang.haro}@upct.es

Abstract. In this paper, we propose and analyze a multicast application called SOMA (SynchrOnous Multicast Application) which offers multicast file transfer service in an asymmetric intra-campus environment. For efficient bandwidth utilization, SOMA uses IP multicasting. We also propose a complete multicast transport protocol involving both, the flow and error correction algorithms. The protocol adapts the window size and the overall application transfer bitrate to the minimum network capacity, allowing synchronism and reacting quickly when congestion arises at any network router. The application behavior has been intensively tested by simulation and experimentally in a lab, using a mixture of wired and wireless intra-campus networks. In addition, we develop a mathematical model to validate analytically some of the most important protocol parameters. The methodology employed to define, analyze and evaluate this multicast protocol is, indeed, another contribution of the work and can be easily extended to other multicast protocols.

Keywords: Multicast, flow and congestion control, transport protocol.

1 Introduction and Related Work

The use of multicasting within a network has many strengths. Multicast minimizes the link bandwidth consumption because no multiple unicast connections are needed to send the information. In addition, it also reduces the sender and router processing and the delivery delay. On the other hand, IP multicasting may be used to add anonymity to a communication, because there is not a univocal relationship between an IP multicast address and a host.

In this paper we propose, analyze, implement and test a SynchrOnous Multicast Application called SOMA to synchronously transfer a large amount of data from a server to a group of clients. It is specially featured to operate in an intra-campus environment (several interconnected LANs through few routers).

Multicast transport protocol requirements (flow and congestion control, error correction, etc.) are more complex than in a point-to-point one. Since TCP is a unicast oriented protocol, it cannot be directly used in a multicast environment.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 703–714, 2006.

Therefore, the choice of an adequate transport protocol is the key issue in the multicast application development.

The extreme complexity associated to the definition of a global multicast transport protocol that meets the requirements of all types of multicast applications leads the designers to several approaches for the transport protocol. The most widespread solution consists of the definition and codification of a specific multicast transport protocol which fits the requirements of an application.

Several multicast transport protocols were proposed to meet the requirements of delay-sensitive, real-time interactive applications, such as RTP/RTCP [1] to support multi-party multimedia conferencing tools, SRM [2] and TRM [3] to support distributed whiteboard tools, etc. These applications can tolerate a certain degree of data loss, but they are sensitive to packet delay variance.

On the other hand, other protocols were proposed to meet the requirements of reliable data distribution services, such as multipoint file transfer. These applications are not delay-sensitive, but require that the information is entirely received, or else the transfer fails. The Muse protocol [4] (which was developed to multicast news articles on the MBone), MDP [5] (the evolution of a protocol used in disseminating satellite images over MBone) and MFTP [6], RMTP [7] and TMTP [8] (other protocols for reliable one-to-many data transmission) are examples of this kind of protocols. Most of them are designed to work in the MBone when the number of receivers is too large (thousands of receivers). To reach scalability and, therefore, to solve the feedback implosion problem, some of them define complex hierarchical topologies and they even introduce some non-layer 3 functionality into the network devices.

In recent years, the IETF Reliable Multicast Transport (RMT) group [9] has taken a different approach to design a set of multicast protocols to suit the variety of applications and service requirements for one-to-many and many-to-many communications. Instead of defining and standardizing multiple protocols, they are defining “building blocks” and two “protocol instantations” [10]. Building blocks are modular components that solve a particular functionality common to multiple protocols. They include, among others, forward error correction schemes, two congestion control algorithms (PGMCC and TFMCC) and generic mechanisms for router assistance. Protocol instantations define how to combine one or more building blocks to create a working protocol. The first one is the Negative-Acknowledgment Oriented Reliable Multicast (NORM), which describes the framework and common components relevant to multicast protocols based primarily on NACK operation for reliable transport. The second one is the Asynchronous Layered Coding (ALC) protocol, which describes a massively scalable reliable content delivery protocol. ALC uses a multiple rate congestion control building block that is feedback free. A sender sends packets in the session to several channels at potentially different rates and receivers just adjust their reception rates individually by joining and leaving channels associated with the session. ALC uses the FEC building block to provide reliability.

Our objective is to define a synchronous multicast transport protocol to be used by our SOMA application in an asymmetric intra-campus environment.

Building blocks proposed by the RMT group are too complex since they cover a general multicast transport scenario. Therefore, we have recovered the first protocol design approach. We propose a complete, compact, and also simple SOMA transport protocol to be used by our SOMA application.

Obviously, our solution requires multicast routing facilities, but this is not a problem since involved routers are located into our administrative domain. In spite of its simplicity, our proposed protocol provides the main tasks of a transport protocol: Efficient and simple flow control, congestion control and error correction algorithms.

SOMA protocol simplicity makes possible an easy codification and a feasible mathematical analysis of the main key features which enables the optimization of some parameter values. It has been written in C language using standard Linux kernel routines.

The paper is organized as follows. Section 2 describes the protocol. Section 3 analytically obtains the key protocol parameters. Section 4 presents our test results in a mixed wired and wireless LAN. Finally, section 5 concludes the paper.

2 SOMA Description

SOMA is a multicast application designed for transmitting synchronously large files and hard disk partitions to a set of clients. This protocol is an extension and enhancement of a previous work [11] to cover asymmetric intra-campus networks. SOMA introduces a transmission window to improve the obtained throughput. We also implement an improved flow control mechanism that allows SOMA to be used when unequal capacity networks are interconnected (asymmetric networks). This is a frequent situation when wireless and wired network coexist. Moreover, in wireless networks (whose proliferation has not doubt, nowadays), the available throughput does not only depend on the number of applications which share the network. In fact, it changes depending on the network capacity, which depends on the signal to noise ratio and other physical parameters. Therefore, it is important to design an adequate flow control mechanism that quickly reacts when congestion arrises.

The application employs IP multicast addressing and implements its own transport protocol over UDP. Thereby, port multiplexing and error checking facilities are automatically resolved by the kernel. However, due to the UDP simplicity, the flow control and error recovery mechanisms have to be implemented to fit the transport layer requirements of our application. For this reason, we alternatively refer to SOMA as an application or as a transport protocol.

2.1 Overall Protocol Description

SOMA splits the transmission process into **two phases**. In the **first one**, the server multicasts a set of data packets (a transmission window) to all clients. The clients store the payload and contend to confirm the received packets by an ACK. Although in this phase the server never retransmits any data packet, a client issues a NACK packet when packet losses are detected and it also saves an error mark instead of the packet payload. The feedback information (ACK and NACK packets) received at the server are used to resize the transmission window. The above procedure is repeated until the file is completely transferred.

The **second phase**, which is focused on error correction, starts when the entire file has been transmitted. Each receiver re-scans its file looking for error marks. If one error is found, the client delivers a unicast Repair-Request packet towards the server. The server answers the client sending a unicast Repair-Response packet.

Error correction tasks are relegated to a final phase since current network technologies offer low error rates. This assumption avoids a complex protocol design, solving infrequent packet losses during the transmission.

One of the main SOMA protocol features is synchronicity. The proposed flow control algorithm, which is explained and tested below, adapts the server transmission rate to the slowest bitrate of a participant network. Therefore, all the clients receive the information at the same time.

SOMA is mainly used to replicate a large amount of information. In this scenario, the reduction of packet flows to only one multicast flow is the objective, and synchronicity is thus, a consequence but not the main concern. However, disabling the error correction phase, the synchronicity feature converts SOMA into a useful and simple multicast transport protocol also for on-line applications.

2.2 Proposed Header

The SOMA packet header consists of 4 fields. The Sequence Number (SN, 4 bytes long) used mainly for packet loss detection. The Type Of Packet (TOP, 1 byte), which distinguishes a DATA, an ACK, a NACK, a Repair Request or a Repair Response packet. The Payload Length (PL, 2 bytes) indicates the total packet length in bytes. The Last Window Sequence Number (LWSN, 4 bytes) is used to indicate the last packet of a given window and then to implement a effective feedback reduction scheme. The header is followed by the payload, which transports 512 information bytes.

2.3 Flow Control Algorithm

After a data packet is sent by the server, it starts a timer called timeout and immediately it waits until an ACK packet for each participating LAN (not for each client) acknowledges the window or until the timer expires. If the timer expires before the ACKs are received, its value is increased multiplying it by a factor of α ($\alpha > 1$). But if the window is confirmed in time, the timer value is decreased as denoted by expression (1)

$$T_{out} = max\{\frac{Tout}{\beta}, default\_Tout\} \tag{1}$$

Where $\beta > \alpha > 1$ and *default_Tout* is the bottom threshold value. The server repeats this operation until the file is completely transferred.

A window is only confirmed when the server receives one ACK for each participating LAN, ensuring synchronism among all multicast clients. Therefore, if one of the networks suffers congestion, the timeout value is increased and therefore, the data transmission rate decreases. When congestion disappears, the timer redefinition allows to increase the transfer rate again.

To improve the flow control reaction, it is convenient that not only the timer but also the window size changes appropriately. To accomplish this, just before sending the next data window, the server modifies the window size as follows:

- If the expected ACKs associated to this window have been received before the timer expires, the server increases the window size in one unit.
- If the timeout expires, the server decreases the window size in one unit.
- For each NACK that indicates a different packet loss (only the first NACK indicating a particular packet loss is considered), the server decrements the window size in one unit.

On the other hand, the clients are waiting for data packets. When a packet arrives, each client extracts the SN and compares it with the expected value:

- If SN is the expected one, the client stores the payload and updates the sequence number.
- If SN is greater, the client detects packet losses and sends a NACK with the sequence number of the received data packet. Simultaneously, it finds out the number of lost packets and it stores an error mark for each one. Finally, it also stores the data contained in the received packet.
- If SN is smaller, the data packet is discarded.

In addition, if the SN matches with the LWSN value, the client competes for sending an ACK to confirm the entire window issued by the server (see the feedback implosion reduction below).

2.4 Feedback Implosion Reduction

To reduce the amount of ACK feedback packets in the network, a client must wait a random period called ARTP (ACK Random Time Period) before sending an ACK and simultaneously, it listens if another client belonging to its LAN is transmitting the same ACK. If the ARTP expires and the ACK has not been received, the client generates and multicasts its own ACK. The rest of clients will receive the ACK but only the clients at the ACK sender side (belonging to the same subnetwork) will disable its own ACK transmission. The ARTP value is obtained from a uniform probability distribution function ranging between zero and $ARTP_{max}$. Thereby, only one ACK for each participant LAN is sent to the server, independently of the number of clients.

The effective ACK generation time is a random variable defined as: $ARTP = min(ARTP_1, \cdots, ARTP_n)$, where n is the number of clients. Therefore, the mean ARTP value is [11]

$$\overline{ARTP} = \left(1 - \frac{n}{n+1}\right) \cdot ARTP_{max} \tag{2}$$

It is clearly decreasing with the number of clients.

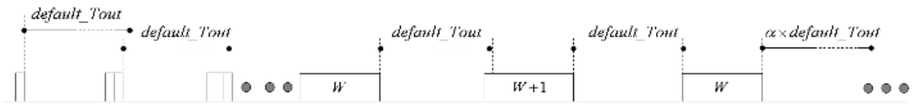


Fig. 1. Window size evolution in an asymmetric network environment

Figure 1 briefly summarizes the usual protocol operation. The server sends a set of data packets, each time increasing the window size until W size is reached. At this point, the timer expires just before all ACKs are received, probably because at some network point congestion arises. The server reacts quickly increasing the timer value and decreasing the window size. It is clear that for protocol consistency, the timeout must be greater than the mean ARTP value ($\overline{ARTP}$).

3 Protocol Characterization

The protocol behavior is strongly correlated with the flow control performance. In particular, the maximum window size, the steady state window size and the maximum throughput values are the three most important protocol parameters.

3.1 Maximum Window Size

The transmission rate is determined by the network capacity, the timeout timer and the window size. The proposed flow control algorithm modifies the last two parameters to reach an optimum transfer rate.

If there is no congestion, the server increases the window size up to its maximum value (supposing also an error-free transmission channel). To simplify, but without loss of generality, it is supposed that there is only one LAN with capacity C bps. Let us also suppose that the file size is large enough to assume that the transmission is performed by the maximum window size. Under these conditions, the total transfer time can be calculated as

$$T = \frac{FileS}{PayloadS} \cdot \frac{DataPS}{C} + \frac{FileS}{PayloadS \cdot W} \left(\overline{ARTP} + \frac{AckPS}{C} \right) \tag{3}$$

Where $FileS$ is the file size, $PayloadS$ is the data packet payload size, $DataPS$ and $AckPS$ are the data and ACK packet sizes respectively, and W is the maximum window size.

The first addend is the time needed for the server to transfer the file and the second one is the average time required by the clients to issue the ACKs. It is obvious that a high maximum window value enables a faster transmission rate, but at the same time the protocol has fewer opportunities to react to network congestion.

By simply operating in (3), the transfer time reduction due to the use of a window size W_2 instead of W_1 ($W_2 > W_1$) is equal to

$$\frac{FileS}{PayloadS}\left(\overline{ARTP}+\frac{AckPS}{C}\right)\cdot\frac{W_2-W_1}{W_1W_2} \tag{4}$$

If an appropriate window size W_1 is selected, an alternative window size W_2 (where $W_2 >> W_1$) does not provide a remarkable transfer time reduction since

$$\lim_{W2\to\infty}\frac{W_2-W_1}{W_2W_1}=\frac{1}{W_1} \tag{5}$$

According to (4) and (5) we choose a maximum window size of 100 data packets (rule of thumb) since it achieves a fast data transmission rate, a quick response when congestion arises, and it avoids protocol starvation (that is, it enables to fairly share the network capacity with other flows).

3.2 Window Size Convergence

The window size during the transmission reaches a steady state value, which is strongly correlated with the throughput. In this section we derive a mathematical expression to this parameter.

In our analytical model, we must assume some simplifications to reduce the extremely complex general situation, which, however, does not invalidate the generality of our analysis. We assume that the intra-campus network consists of unequal capacity LAN networks (some of them working at C_1 and the others at C_2, where $C_1 >> C_2$) connected through multicast routers. We also assume that there are no other applications using the network and that the server is reasonably situated at one of the fastest LANs.

Congestion may arise in routers interconnecting LANs with different capacities. Those routers can be modeled as a pair of buffers serving packets at C_1 and C_2 Mbps respectively.

Supposing an initial window size of one (see figure 1), the server sends only one data packet to the network and it waits for ACKs (one from each LAN). The last ACK received at the server is the ACK going through the path formed by the highest number of C_2 networks (it is composed by N_{C1} LANs at C_1 Mbps networks and by N_{C2} LANs at C_2 Mbps). When all ACKs have arrived, the window size is increased by one unit and the next data window is issued. For a W window size, the server will receive the last ACK approximately at

$$\frac{W\cdot DataPS}{C_2}+(N_{C2}-1)\frac{DataPS}{C_2}+N_{C1}\frac{DataPS}{C_1}+\overline{ARTP}+N_{C1}\frac{AckPS}{C_1}+$$
$$+N_{C2}\frac{AckPS}{C_2}\approx\frac{W\cdot DataPS}{C_2}+(N_{C2}-1)\frac{DataPS}{C_2}+\overline{ARTP} \tag{6}$$

where LAN_2 to LAN_1 buffer delay can be neglected because the service rate at the other side is very high (C_1 Mbps).

The window size just before congestion is detected (W_T) can be obtained when (6) slightly matches with *default_Tout*:

$$W_T = \left\lfloor \frac{\left(default\_Tout - \overline{ARTP}\right)}{DataPS} \cdot C_2 - (N_{C2} - 1) \right\rfloor \tag{7}$$

It can be noticed that for each C_2 network added to the critical path, the W_T value is decremented in one unit.

At this time, the server increases the window size again and it sends the next data block. Now, congestion is declared since the timer expires before the last ACK packet arrives. Therefore, the flow control multiplies the timer by α and decreases the window in one unit. In this new situation, it can be guaranteed that the server assumes the congestion has disappeared, since $\alpha > 1$. Once again, the window is increased and the timer is divided by β. But since $\beta > \alpha > 1$, the timer value reaches its default value again and then congestion comes back. This behavior is continuously repeated. Therefore, the window size reaches a steady-state value slightly oscillating around W_T.

3.3 Maximum Throughput

SOMA obtains the maximum throughput and the maximum window size (W_{max}) when it is the only running application and there is no congestion. In that situation, the time interval between two consecutive data windows is restricted by the ARTP mean value (2) and not by the timer ($default\_Tout >> ARTP_{max}$). Therefore, in this case the maximum throughput is bounded by

$$\frac{W_{max} \cdot DataPS}{\frac{W_{max} \cdot DataPS}{C} + \overline{ARTP}} \tag{8}$$

Where C is the network capacity in bps at the server side.

However, if congestion arises at some network point, the timeout timer restricts the time between data blocks and the window size reaches its steady-state value. Therefore, the maximum throughput is bounded by

$$\frac{(W_T + 1) \cdot DataPS}{\frac{(W_T + 1) \cdot DataPS}{C} + default\_Tout} \tag{9}$$

4 Test Results Discussion

In this section, we evaluate SOMA in a real situation. It should be noticed that our analytical study is focused on a transport layer but test experiments are obviously the result of all OSI layers integration, from the physical layer up to the transport one. Particularly, in section 3 we have not taken into consideration the MAC, LLC, IP and UDP protocols and sub-layers. Moreover, SOMA runs over a multi-task OS, which has non real-time facilities (Linux kernel 2.4). Therefore,

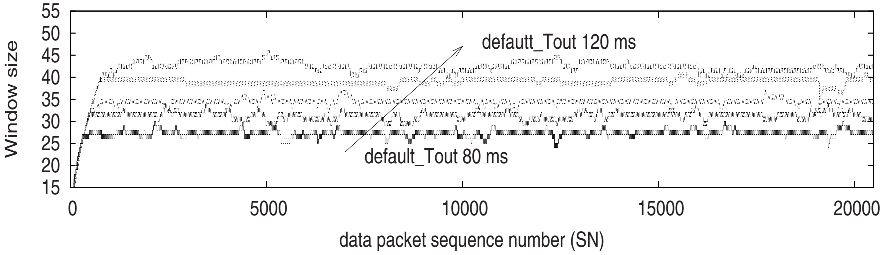


Fig. 2. Window size evolution for different *default_Tout* values: 80, 90, 100, 110 and 120 ms

although we try to minimize the computational load in each computer (unnecessary processes, like *cron*, are killed), sometimes the kernel may give priority to other processes instead of SOMA. Both effects, the OSI layers integration and the multi-task OS may cause that the test results reveal some smaller differences with the analytical ones.

The intra-campus environment is formed by two LANs of extremely unequal capacities, a wired Ethernet LAN at 100 Mbps and a wireless LAN 802.11b at 2 Mbps, both connected through a wireless access-point router. The access-point router is a Linksys WRT54G, co-sponsored by Cisco Systems. We changed its firmware by a stable and configurable Linux OS called OpenWrt [12].

To verify that the analytical results obtained in section 3 fit well enough with the test results, the same intra-campus environment is used: the clients are situated in both LANs and the server is situated in the wired network.

Our test intra-campus network forces congestion since the wireless LAN capacity (2 Mbps) is fifty times lower than the wired network capacity (100 Mbps).

Figure 2 shows the evolution of the window size for different *default_Tout* values: 80, 90, 100, 110 and 120 ms. According to expression (7), the window size should oscillate around 29, 32, 36, 39 and 43 packets respectively. To obtain these values it is assumed that: (a) The $\overline{ARTP}$ is 120 μs, which is calculated using (2) when n=4 and the $ARTP_{max}$ is 600 μs. (b) The effective wireless LAN capacity at the transport layer is around 1.55 Mbps instead of the theoretical 2 Mbps due to the OSI layers integration.

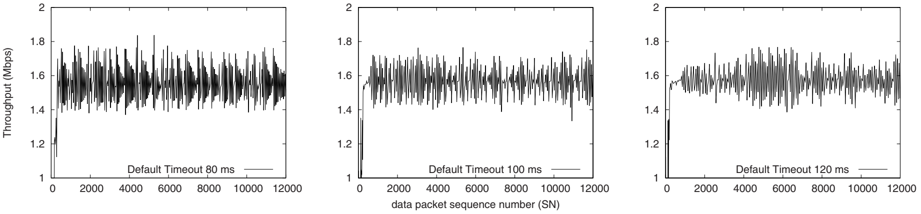


Fig. 3. Instantaneous throughput evolution for different *default_Tout* values: 80, 100 and 120 ms

Table 1. W_T values obtained theoretically and by simulation (in parenthesis), supposing a wireless LAN capacity C_2 of 1.55 Mbps and $\overline{ARTP}$=120 μs

	default_Tout				
N_{C2}	80 ms	90 ms	100 ms	110 ms	120 ms
1	29 (29)	33 (33)	37 (37)	40 (40)	44 (44)
2	28 (28)	32 (32)	36 (36)	39 (39)	43 (43)
3	27 (27)	31 (31)	35 (35)	38 (38)	42 (42)
4	26 (26)	30 (30)	34 (33)	37 (37)	41 (41)

As it can be observed, the analytical values fit well enough with the experimental ones and the window size always remains around its steady state value (W_T). Sometimes the window size slightly decreases due to sporadic packet losses at the wireless LAN side and also because of background control applications packets, such as BPDU spanning-tree, which overload the access point buffer capacity.

Additionally, we have validated our window size convergence study in more complex scenarios using the Opnet simulator. Each possible scenario is formed by several C_1 and C_2 networks so that the last ACK packet received at the server goes through a path formed by N_{C1} and N_{C2} networks. Table 1 presents the W_T value obtained by simulation and theoretically (7) when the value N_{C2} varies among 1 and 4.

It can be observed that simulated results validate the analytical study. In addition, the case $N_{C2} = 1$ (the scenario studied experimentally) fits good enough with the experimental results showed in figure 2.

Returning to test experiments, figure 3 represents the instantaneous throughput. Irrespective of the *default_Tout* value, the server throughput slightly oscillates around 1.55 Mbps. Therefore, the proposed flow control algorithm is able to adapt the server transmission rate to the slowest network capacity using a unique flow, maintaining synchronism among all clients and avoiding congestion.

This test result can be corroborated analytically by introducing the value of W_T (7) in (9) when $N_{C2} = 1$. Always assuming that mean ARTP value is negligible, the throughput can be approximated by

$$\frac{default\_Tout \cdot C_2 + DataPS}{\dfrac{default\_Tout \cdot C_2 + DataPS}{C_1} + default\_Tout} \approx C_2. \tag{10}$$

Where $C_2 << C_1$ and $DataPS << default\_Tout \cdot C_2$

In the next experiment, our protocol is evaluated in a single congestionless wired LAN. In this scenario the window size reaches its maximum value limited by the protocol (W=100) and the maximum experimental throughput is around 97 Mbps, which approximately matches the theoretical result (97.4 Mbps, from equation 8). Again, the flow control is able to adapt the transmission to the maximum network capacity.

Finally, figure 4 illustrates the window size evolution in a different experiment. At the beginning only wired clients participate in the file replication process.

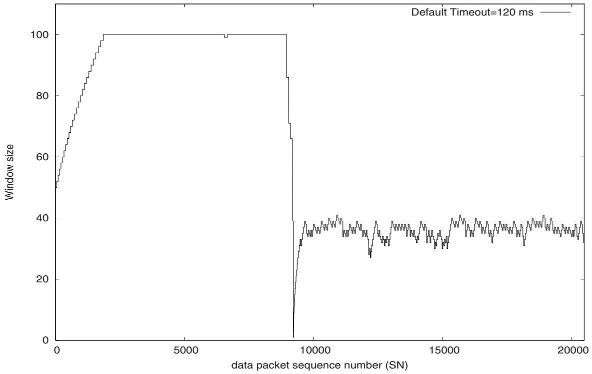


Fig. 4. Window size evolution in a mixed wired and wireless intra-campus. The wireless LAN terminals join the file transfer approximately in the middle of the transfer.

As it can be seen, the window size reaches its maximum value (W=100). But approximately in the middle of the transfer, the wireless terminals join the file transfer. As it can be appreciated, the SOMA flow control is able to quickly adapt to the new situation by resizing the window (and also the timer, although it is not shown) synchronizing both networks and avoiding congestion. If the router buffer is not high enough, some data packets could be lost during the transition period, which will be recovered in the error correction phase. To minimize this effect, the response time of our proposed protocol is an important factor since the wireless channel capacity is strongly dependent on physical parameters.

5 Conclusions

SOMA is a multicast application for fast file replication. One of its most remarkable aspects is its own transport protocol definition focused mainly on flow control which is designed to work fine in an asymmetric intra-campus scenario. The proposed flow control algorithm is able to quickly react under congestion, resizing adequately the window size and the time between data blocks to maximize the throughput.

Some of the main protocol parameters have also been characterized analytically under certain constrains. In addition, the mathematical study has been validated with real traces in a test lab network.

Although the proposed transport protocol is used in SOMA for file transfer, its synchronicity and simplicity makes it interesting for other type of applications, like on-line applications.

Acknowledgments

This work has been supported by the Spanish Research Council under project ARPaq (TEC2004-05622-C04-02/TCM).

References

1. Schulzrinne, H. et al.: RTP. A Transport Protocol for Real-Time Applications. RFC 3550, Internet Engineering Task Force, July 2003.
2. Floyd, S., Jacobson, V., Liu, C., McCanne, S., Zhang, L.: A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. IEEE/ACM Transactions on Networking, Vol. 5, No. 6, pp. 784-803, December 1997.
3. Sabata, B., Brown, M. J., Denny, B. A., Heo, C.: Transport protocol for reliable multicast: TRM. In Proc. of IASTED International Conference on Networks, pp. 143-145, January 1996, Orlando, Florida.
4. Lind, K. et al.: Drinking from the Firehose: Multicast USENET News. In Proc. of the Winter 1994 USENIX Conference, pp. 33-45, 1994, San Francisco, CA.
5. Macker, J.: The Multicast Dissemination Protocol (MDP) Toolkit. In Proc. of IEEE MILCOM, Vol. 1, pp. 626-630, 1999.
6. Miller, K. et al.: StarBurst Multicast File Transfer Protocol (MFTP) Specification. IETF Internet Draft, draft-miller-mftp-spec-03.txt, April 1998.
7. Lin, J.C., Paul, S.: RMTP. A Reliable Multicast Transport Protocol, In Proc. of Infocom96, pp. 1414-1424, March 1996, San Francisco, CA.
8. Yavatkar, R. et al.: A reliable dissemination protocol for interactive collaborative applications. In Proc. of the ACM Multimedia'95, pp. 333-344, 1995.
9. http://www.ietf.org/html.charters/rmt-charter.html
10. Kermode, R., Vicisano, L.: Author Guidelines for RMT Building Blocks and Protocol Instantiation documents. IETF Internet Draft, draft-ietf-rmt-author-guidelines-03.txt, January 2002.
11. Manzanares-Lopez P., Sanchez-Aarnoutse J.C., Malgosa-Sanahuja J., Garcia-Haro J.: Empirical and Analytical Study of a Multicast Synchronous Transport Protocol for Intra-Campus Replications Services. In Proc. of the International Conference on Communications (ICC'04), June 2004, Paris, France.
12. http://openwrt.org

Multi-Layer Traffic Engineering Through Adaptive λ-Path Fragmentation and De-fragmentation[*]

Tibor Cinkler, Péter Hegyi, Márk Asztalos,
Géza Geleji, János Szigeti, and András Kern

HSN*Lab*, Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics,
Magyar tudósok körútja 2, H-1117 Budapest, Hungary
{cinkler, hegyi, asztalos, geleji, szigeti, kern}@tmit.bme.hu

Abstract. In Multi-Layer networks, where more than one layer is dynamic, i.e., connections are set up using not only the upper, e.g., IP layer but the underlying wavelength layer as well leads often to suboptimal performance due to long wavelength paths, that do not allow routing the traffic along the shortest path. The role of MLTE (Multi-Layer Traffic Engineering) is to cut these long wavelength paths into parts (fragments) that allow better routing at the upper layer (fragmentation), or to concatenate two or more fragments into longer paths (defragmentation) when the network load is low and therefore less hops are preferred.

In this paper we present a new model (GG: Grooming Graph) and an algorithm for this model that supports Fragmentation and De-Fragmentation of wavelength paths making the network always instantly adapt to changing traffic conditions. We introduce the notion of *shadow capacities* to model "lightpath tailoring". We implicitly assume that the wavelength paths carry such, e.g., IP traffic that can be interrupted for a few microseconds and that even allows minor packet reordering.

To show the superior performance of our approach in various network and traffic conditions we have carried out an intensive simulation study.

Keywords: Adaptive Multi-Layer Traffic Engineering, Grooming Graph, Wavelength Path Fragmentation and De-Fragmentation.

1 Introduction

The evolution of transport networks shows two main directions. First, there are multiple networking technologies layered one over the other. Second, it is required that not only the upper-most layer is dynamic, i.e., switched, but the upper two, or maybe all the layers.

If the layers of this vertical structure are run by different operators or providers then they must communicate to each other to exchange information necessary

[*] This work has been done as a part of the European FP6 IP NOBEL (www.ist-nobel.org) and NoE e-Photon/ONe (www.e-photon-one.org) research projects.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 715–726, 2006.

for routing and other purposes. This vertical communication is referred to as Interconnection, and there are three defined Interconnection Models: (1) Overlay, (2) Augmented and (3) Peer model [1].

If all these layers are run by a single operator or provider then there is no need for communication interfaces between the layers. Therefore, a single unified integrated CP can be used for all the layers and then we have instead of the interconnection the so called *Integrated Model.* The forwarding units of all the layers of the data plane are connected to a single control plane unit.

Similarly, if such a Multi-Layer network has layers or some parts of certain layers built of interconnected elements of a unique networking technology then the set of these elements is referred to as a *Region.* Having multiple different regions within a network is referred to as a *Multi-Region* network [1] [2].

In switched multilayer transport networks (e.g. ASTN/GMPLS) the traffic demands have typically bandwidth by orders of magnitude lower than the capacity of λ-links. Therefore, it is not worth assigning exclusive end-to-end λ-paths to these demands, i.e., *sub-λ granularity* is required. Furthermore, the number of λs per fibre is limited and costy. To increase the throughput of a network with limited number of λs per fibre *traffic grooming* capability is required in certain nodes. There are many papers dealing with routing, traffic engineering and resilience in such multilayer networks, where grooming is one of the key issues [2] [3] [4] [5].

Here we consider the case of Wavelength Routing Dense Wavelength Division Multiplexing (WR-DWDM) Networks and one layer built over it. In the WR-DWDM layer a wavelength path (λ-path) connects two physically adjacent or distant nodes. These two physical nodes will seem adjacent for the upper layer built over it.

This upper layer is an "electronic" one, i.e., it can perform multiplexing different traffic streams into a single λ-path via simultaneous time and space switching. Similarly it can demultiplex different traffic streams of a single λ-path. Furthermore, it can perform re-multiplexing as well: Some of the demands demultiplexed can be again multiplexed into some λ-paths and handled together along it. This is referred to as *traffic grooming* [6] [3]. Further on we will refer to it as *grooming.* This electronic layer is required for multiplexing packets coming from different ports (asynchronous time division multiplexing). It can be a classical or "next generation" SDH/SONET, MPLS, ATM, GbE, 10 GbE or it can be based on any other technology. However, in all cases the network carries mostly IP traffic. The only requirement is that it must be unique for all traffic streams that have to be de-multiplexed, and then multiplexed again, since we cannot multiplex e.g. ATM cells with Ethernet frames directly.

More generally, we can consider this two-layer approach as two layers of a 4-5 layer GMPLS/ASTN architecture [7] [8] [9]. However, not only the framing and layering structure is of interest, but also the control plane proposed in the GMPLS/ASTN framework.

Many excellent papers deal with design, configuration and optimisation of WDM Networks. Some of these methods can be generalised for on-line routing in two-layer networks as well using the model we propose in this paper.

There are also numerous papers dealing with on-line routing in WR-DWDM networks (see, e.g., Chapter 3 of [10]). There are multiple papers on grooming, mostly for the static case, i.e, when a two layer network is configured (see, e.g., Chapter 4 of [10]). Some papers consider the grooming capability in dynamic (on-line) routing [11]. There are also papers dealing with multilayer survivability, e.g., [12] and Chapter 5 of [10]).

However, there are only few papers, e.g., [4] [13] [14], that take all these into account *simultaneously*, using the peer or the MRN model. To our knowledge there is no paper that proposes any method for adaptive, automated, on-line and distributed Multi-Layer Traffic Engineering. The aim of our paper is to fill this gap.

Our objective is to perform distributed on-line routing of the on-line arriving demands with estimated effective bandwidths as constant bandwidth pipes over the two network-layers optimally in distributed way without separating these layers. The upper layer is assumed to support multiplexing (e.g., asynchronous TDM), while the lower layer is the λ-path system. Separating the two layers decreases the complexity, however, it also deteriorates the routing. According to the role of TE (Traffic Engineering) to increase throughput while it maintains QoS, grooming is performed jointly with adaptive, on-line, distributed and automated path fragmentation and defragmentation.

The rest of the paper is organised as follows. In Section 2 we present the "Grooming-Graph" and the "Shadow-Capacities" and explain how our model works with available routing protocols. In Section 3 the problem of λ-path fragmentation and defragmentation is explained and in Section 4 the simulation results are shown and discussed.

2 The Grooming Graph (GG) Model

The objective was to provide a general network model for routing in two layer networks with grooming, with different types of nodes and arbitrary topologies assuming peer/MRN-model, that allows optimal routing, using the resources of both layers jointly. The aim was to allow adaptive, automated, distributed MLTE (Multi-Layer Traffic Engineering) by the model used.

Although the most widespread topology is ring or interconnected rings, the model must be able to handle any regular or mesh topology. Furthermore, it must be able to handle any type of nodes of practical interest, e.g., OADM, OXC, EOXC, etc., all with or without grooming capability and with or without λ-conversion capability. Even limited grooming, and λ-conversion limited in number or range has to be supported. For this purpose we use our grooming graph (GG) model, where the node is substituted by a sub-graph.

The simpler version of this model that does not allow fragmentation and defragmentation was first proposed in [15]. ILP formulation of the static RWA problem with grooming and protection was given in [16], using the wavelength graph, while in [17] heuristics for solving the problem were proposed. [18] explains the used simpler model the "WG" and investigates the fairness issues of dynamic grooming.

In this paper we add the adaptivity to the model by defining the shadow links and their shadow capacities to be used for adaptive MLTE.

2.1 Model of Links

A network consists of nodes, and links connecting the nodes. This can be modelled by a graph: a node is a vertex and a link is an edge. Having multiple λs (WL1-WL3) we will represent a λ of a link as an edge in the graph of wavelengths, according to Figure 1 for the network proposed in [19]. To prioritise filling up λs one-by-one we can assign slightly different weights to different λ-channels of one link. For example, edges representing WL1, WL2 and WL3 in Figure 1 will have weights 1,01, 1,02 and 1,03 respectively.

2.2 Model of Nodes

A node is modelled by a subgraph. The subgraph-nodes are the switch-ports, while the weighted edges represent the costs of transitions, terminations, conversions, etc. There are different types of nodes. Models of nodes differ for these. Some examples will be shown here. In similar manner a model can be derived for any additional node-type.

Optical Add-and-Drop Multiplexer (OADM): The OADM Nodes have in general two bi-directional ports (4 fibres). Their function is either to transmit a λ-path or to terminate it and usually they do not allow λ-conversion.

The weights assigned to edges representing termination (e.g., 50) are higher than weights of transition (e.g., 25), because transition is preferred to termination. According to the proposed model (Figure 2) the traffic streams can either enter or exit the OADM crossing vertex E or can be even re-multiplexed.

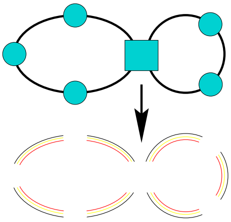


Fig. 1. Modelling edges in the GG

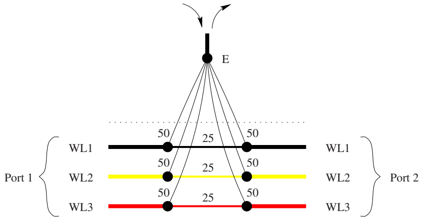


Fig. 2. Model of OADM nodes

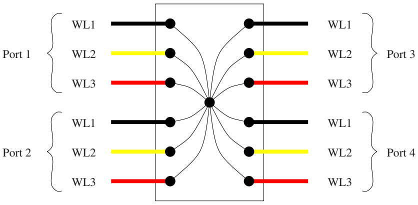


Fig. 3. Model of EOXC nodes

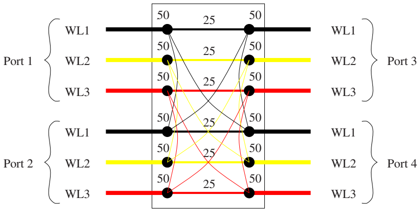


Fig. 4. Simple OXC (no λ-conv.)

Cross-Connect with Electronic Core (EOXC): In the model shown in Figure 3 each pair of nodes should be connected by an edge, representing potential Cross-Connection. All edges should have equal weights. Instead of connecting all pairs using nxn edges we use n edges and one node. This simplifies the model. Each incoming channel is converted to electrical domain switched by a space-switch and again converted to the optical domain to arbitrary λ. Each termination, transition or λ-change has the same cost (e.g., 25). Therefore all edges have the same weight (e.g., 25/2).

Optical Cross-Connect (OXC): An optical Cross-Connect has more than two ports, e.g., four bi-directional ports according to Figure 1. In an OXC a light-path can make transition to any output port which supports that λ , and that λ is not yet used. This OXC type (without λ-conversion capability) will be referred to as *simple* OXC (see Figure 4). In this case one incoming channel can exit at any of the remaining output ports where that λ is supported and not yet used.

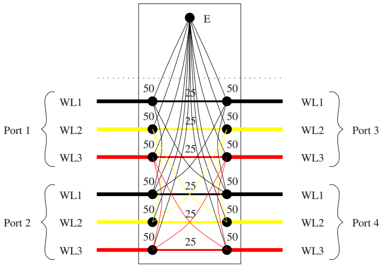


Fig. 5. OXC with λ-conversion

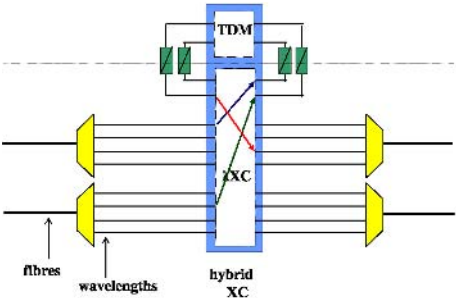


Fig. 6. The GG node model

In some cases the traffic stream termination is also among the functions of an OXC. In that case the model does not need any change. The only difference will be that there will be some traffic offered to that OXC node. This can be modelled by offering traffic to node E and considering it as an end-node. In this case traffic-stream re-multiplexing capability is also required.

Modelling Grooming: Grooming can be modelled analogously to λ-conversion. The difference is that while in case of λ-conversion an incoming traffic stream can exit as a single outgoing stream at another λ, in case of grooming traffic streams can be multiplexed, i.e., instead of space switching space AND time switching/cross-connecting is performed.

These two functionalities can be combined as well within a single model.

Note, that λ-conversion is a special case of grooming. Therefore a node supporting only grooming can perform λ-conversion as well, while a λ-conversion node can not perform grooming.

The model we presented in Section 2 will be referred to as 'simple' grooming model. To make this model better adapt to the traffic and network conditions we extend it in Section 3.

3 Shadow Capacities for λ-Path Fragmentation and De-Fragmentation

We assume either the peer interconnection model or the vertically integrated multi-region network (MRN) node model for multi-layer networks [2]. Then the network layers set the resources jointly, i.e., the control plane has knowledge of both the layers to best accommodate the arriving traffic demands.

This often leads to suboptimal performance, since the lightpaths will be routed depending on the arrival order of demands as well as on the load of the network. For instance in an empty network each arriving demand will be routed over an exclusive lightpath. This will result in a set of long lightpaths that will hinder routing the new demands, i.e., the network will become de-fragmented. After the transients the lightpaths will be configured more or less adequately. However, if the level of traffic grows short lightpaths with plenty of grooming are needed to accommodate it, i.e., lightpaths have to be fragmented into shorter parts.

To have always optimal performance the lightpath system has to adapt to the changing traffic conditions. Unfortunately, in the simple model the virtual topology offered by the wavelength system may not be changed until there is any traffic within the considered λ-paths.

3.1 Algorithm for Routing over Shadow Capacities

Figures 6 - 10 explain the use of shadow links and shadow capacities. Let us consider an example. Figure 6 shows a peer/MRN node that has two incoming and two outgoing fibres each carrying three λs. The bottom part is a wavelength

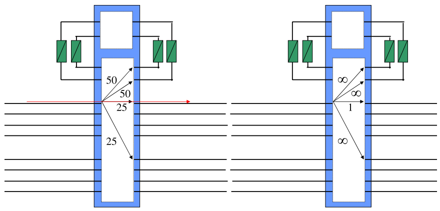


Fig. 7. Setting weights in the GG

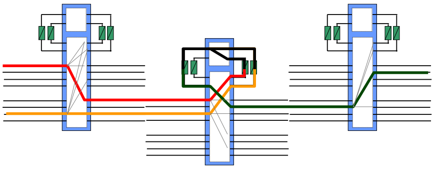


Fig. 8. Routing with grooming

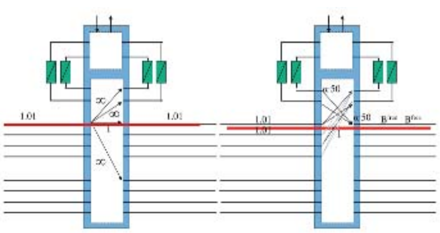


Fig. 9. Creating a "shadow link" of "shadow capacity" of B^{free}

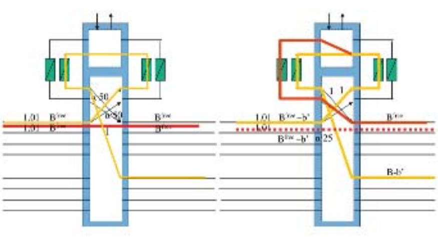


Fig. 10. If routing over the shadow capacities, the λ-paths will be cut

cross-connect, that has two E/O and two O/E converters that connect to the electronic part of the node. In the upper part (marked as 'TDM') the signals can be groomed (or added, or dropped). The figure shows, that the content of two λ-paths is groomed into a single one.

Now, let us see the model of this node (Figure 7). On the left hand side part of the figure we show an example for setting up the internal link weights to be used for routing. Wavelength transition is cheaper (25 cost units) than using the electronic layer, that will cost at least $50 + 50 = 100$ cost units.

Based on these weights set for all the internal and external links in the network model we search for a shortest path between certain nodes. In the right-hand side part of Figure 7 we have chosen a transition, while Figure 8 shows a grooming. Routing is always followed by re-setting the link weights. The right-hand side part of Figure 7 shows the approach used for the simple grooming model, while Figures 9 and 10 introduce the shadow links.

Figure 9 shows that after routing a demand using any of the shortest path algorithms (e.g., Dijkstra's), the internal links connected to internal nodes used by that demand will neither be deleted, nor will be their costs increased to infinity, but increased enough to avoid using those links until other wavelengths or other paths exist. In figure we have multiplied the weights of these links by parameter $\alpha >> 1$. It means that the model allows not only the used internal link, but introduces more expensive shadow links having as much shadow capacity as the free capacity of the internal link used by the considered demand is. This does not mean branching the optical signal, but it gives opportunity to choose instead of the current internal optical link cutting (fragmenting) the λ-path and going to the upper, electronic grooming layer.

Figure 10 shows such a case when there was no cheap alternative wavelength or path, and the more expensive shadow link of the GG had to be chosen while searching for the shortest path that resulted in cutting the λ-path. The right-hand side of the figure shows that the two traffic streams are now demultiplexed (de-groomed), a new shadow link with new shadow capacity has been defined (dotted thick line).

Until there is any free capacity in the λ-paths, they will have shadow links of shadow capacities equal to the free capacity.

In the upper example, we have shown how a λ-path can be cut for grooming purposes. Similarly, if a λ-path does not carry any traffic, it will be cut into λ-links, and the capacity and weight values of these links will be set to their initial values. We refer to this action as *λ-path fragmentation.*

Similarly, two λ-paths can be concatenated if they use the same wavelength AND they are connected to the same grooming node, but there is no third traffic that has to be added or dropped. Although it happens rarely, it is very useful in case when the number of grooming ports is the scarce resource. We refer to this action as *λ-path defragmentation.*

In the remaining part of this paper based on simulation results we show what parameters influence and how do they influence the performance and dynamic behaviour of the network. The blocking was in all cases the lowest for this adaptive grooming approach with λ-path fragmentation and de-fragmentation.

4 Numerical Results

The code was written in C++ under Linux and Windows operating systems, while the simulations were carried out on a Linux MSI K7Dual AMD Athlon 2000+ MP workstation with 2 GBytes of RAM, 2.4.18 kernel. We have applied DES (Discrete Event Simulation) where we route the demands in the given order, however, to speed up the simulation we do not wait between two demands as the time stamps determine, but route the next demand as soon as the last demand is routed.

The test networks were the COST 266 European reference Network [20] consisting of 25 nodes and 32 physical links shown in Figure 11 and the NSFnet consisting of 14 nodes and 21 links shown in Figure 12. We have used OADMs in all nodes of degree 2 and OXCs with grooming capability in all other nodes. We have compared the behaviour of three network node models:

- OXC: Optical cross-connect with no wavelength-conversion capability and no grooming capability.
- OGS: OXC with gromming capability. This is the *simple* grooming node model that we proposed earlier and was used by other authors as well.
- OGT: OXC with grooming capability with support for *"tailoring"* λ-paths, i.e., adaptive, distributed fragmentation and de-fragmentation of λ-paths. This is our new method proposed in this paper.

We investigate how the blocking ratio depends on three parameters, namely the bandwidth of demands, the holding time of sessions and the number of λs per link.

We have assumed 6 wavelengths per link, 1000 units of capacity for all wavelengths, 100 units of bandwidth on average and 8 units of holding time for the demands as the default values, for both, COST266 and NSF networks. Session arrival rate was 0.025 for the COST266 while it was 0.08 for the NSF network.

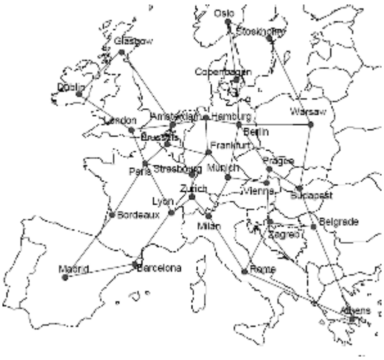


Fig. 11. Basic COST266 topology

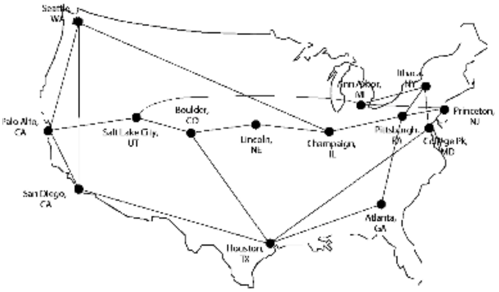


Fig. 12. The NSFnet topology

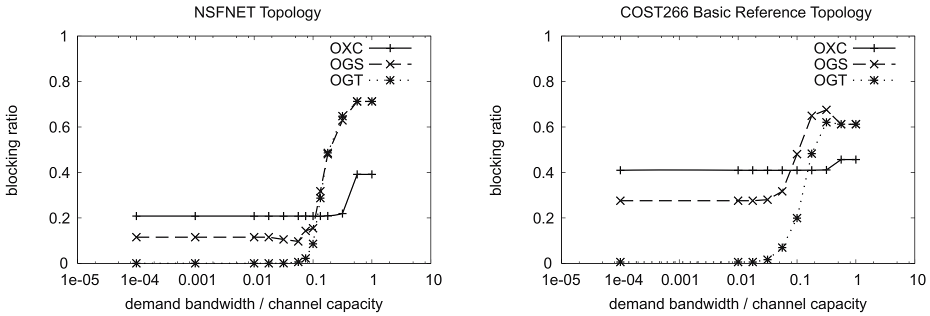


Fig. 13. Blocking ratio vs. the ratio of the demand bandwidth to the channel capacity

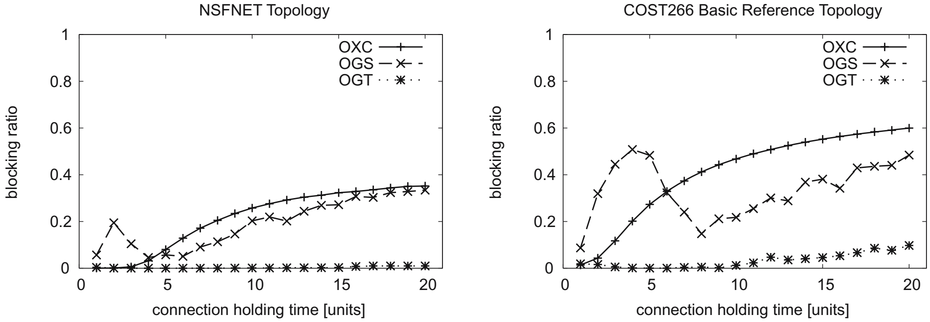


Fig. 14. Blocking ratio vs. the average connection holding time

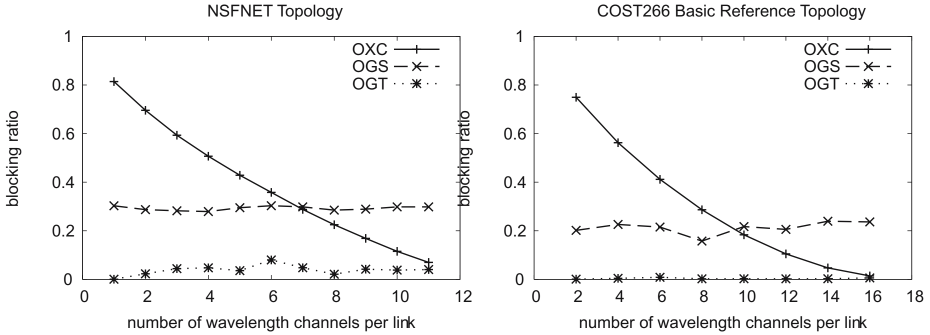


Fig. 15. Blocking ratio vs. the number of wavelengths

As a reference the OXC case was used, i.e., all nodes were OXCs without λ-conversion capability. In this case all the traffic demands have used exclusive λ-paths.

Bandwidth of Demands: First we tune the ratio of the average bandwidth of the demands to the capacity of λ-links (Figure 13). While the bandwidth ratio is significant, there is a huge difference in blocking. Adaptive grooming is superior to simple grooming. However, as the bandwidth ratio approaches 0,1 the blocking grows for both grooming approaches and they become comparable.

It is interesting to note, that blocking of both grooming approaches is larger than that of the approach with no grooming (OXC) as the bandwidth of the demands approaches the capacity of the λ-links. It is probably resulted by the long λ-paths that hinder routing demands over shorter paths. Note, that in our adaptive grooming framework we do not allow rerouting existing connections to other paths, but just cutting or concatenating the λ-path fragments they use for three reasons. Namely, to simplify the operation, keep the adaptive and automatic traffic engineering local, and to keep the interruption time very short.

However, in practice the typical operational region of networks falls out of this critical region, i.e., the typical bandwidth of demands is lower at least by one to two orders of magnitude than the capacity of λ-links.

Holding Time of Demands: Figure 14 shows that when increasing the holding time of connections the blocking grows. Our adaptive grooming approach (OGT) has significantly lower blocking than the other two methods, particularly for the NSF network (Figure 14). It is very interesting that simple grooming (OGS) has higher blocking for short holding times than in the case with no grooming at all (OXC)!

Number of Wavelengths: Figure 15 shows, that increasing the number of λs per link the blocking smoothly drops for the case with no grooming (OXC). The adaptive grooming model (OGT) has always better performance than the other two methods. Both grooming models have roughly the same blocking when the number of λs grow, while the performance of the model with no grooming improves. For large number of λs the simple grooming approach (OGS) has higher blocking than that with no grooming at all! The proposed grooming method has always the best performance. The curves for OXC are very smooth, while for grooming they fluctuate. This supports that grooming inherently introduces numerous anomalies.

5 Conclusion

In this paper we have proposed a new model, the Grooming Graph, that supports distributed, automatic, adaptive and on-line multi-layer traffic engineering performed through adaptive grooming using the *shadow links* and their *shadow capacities*. This approach allows the network to adapt well to changing traffic conditions. The λ-paths are *fragmented* and *de-fragmented* as the network and traffic conditions require in a fully automated, adaptive and distributed way without any centralised action or initialisation while simply using the available routing protocols!

The results show, that our approach yields the lowest blocking ratio in all cases for all scenarios studied. In some cases the blocking is by orders of magnitude lower than that achieved by known methods. Applying the proposed method in networks the throughput can be significantly increased and therefore the revenue as well, while minor investments are needed to upgrade to using this method.

The only limitation of the proposed approach is that separate wavelengths should be allocated for traffic that is sensitive even to these very short interrupts and delay variations needed for λ-path fragmentation and de-fragmentation.

References

1. E. Dotaro, M. Vigoureux, D. Papadimitriou: *"Multi-Region Networks: Generalized Multi-Protocol Label Switching (GMPLS) as Enabler for Vertical Integration"*, Alcatel Technology White Paper, February 2005, researchlibrary.theserverside.net/detail/RES/1109006898_409.html
2. M. Vigoureux, B. Berde, L. Andersson, T. Cinler, L. Levrau, D. Colle, J.F. Palacios, M. Jager: *"Multi-Layer Traffic Engineering for GMPLS-enabled Networks"*, *IEEE Communications Magazine*, July 2004, Vol.x, No.x, pp. x-x
3. E. Modiano, P.J. Lin: *"Traffic Grooming in WDM Networks"*, *IEEE Communications Magazine*, vol.39 no.7 pp.124-129, July 2001
4. T. Cinkler, Cs. Gáspár: "Fairness Issues of Routing with Grooming and Shared Protection", ONDM 2004, 8th Conference on Optical Network Design and Modelling, Ghent, Belgium, February 2-4, 2004
5. M. Perényi, J. Breuer, T. Cinkler, Cs. Gáspár: "Grooming Node Placement in Multilayer Networks", ONDM 2005, 9th Conference on Optical Network Design and Modelling, pp. 413-420, Milano, Italy, February 7-9 2005
6. T. Cinkler: *Traffic and λ Grooming"*, *IEEE Network*, March/April 2003, Vol.17, No.2, pp.16-21
7. A. Banerjee et al.: *"Generalized Multiprotocol Label Switching: An Overview of Signalling Enhancements and Recovery Techniques"*, *IEEE Communications Magazine*, pp. 144-151, July 2001, Vol. 39, No. 7
8. B. Rajagopalan et al.: *"IP over Optical Networks: Architectural Aspects"*, *IEEE Communications Magazine*, pp. 94-102, September 2000, Vol. 38, No. 9
9. R. Sabella, H. Zhang, eds.: *Traffic Engineering in Optical Networks"*, *IEEE Network*, March/April 2003, Vol.17, No.2
10. T. Cinkler et al. eds.: *Proceedings of ONDM 2003*, vol.1, ISBN-963206406 (http://www.hsnlab.hu/˜ ONDM2003)
11. K. Zhu, H. Zhu, B. Mukherjee: *"Traffic Engineering in Multigranularity Heterogeneous Optical WDM Mesh Networks through Dynamic Traffic Grooming"*, *IEEE Network*, March/April 2003, Vol.17, No.2, pp.8-15
12. S. De Maesschalck, et al.: *"Intelligent Optical Networking for Multilayer Survivability"*, *IEEE Communications Magazine*, January 2002, Vol.40, No.1, pp. 42-49
13. H. Zhu, H. Zang, K. Zhu, B. Mukherjee: *A Novel Generic Graph Model for Traffic Grooming in Heterogeneous WDM Mesh Networks*, *IEEE/ACM ToN: Transactions on Networking*, Vol.11, No.2, pp. 285-299, April 2003
14. C. Ou, K. Zhu, H. Zang, L.H. Sahasrabuddhe, B. Mukherjee: *Traffic Grooming for Survivable WDM Networks - Shared Protection*, IEEE JSAC: Journal on Selected Areas in Communications, Vol.21, No.9, pp. 1367-1382, November 2003
15. T. Cinkler, R. Castro, S. Johansson: *"Configuration and Re-Configuration of WDM Networks"*, NOC'98, European Conference on Networks and Optical Communications, Manchester, UK, June 1998

16. T. Cinkler: *“ILP Formulation of Grooming over Wavelength Routing with Protection”*, IFIP ONDM 2001, 5th Conference on Optical Network Design and Modeling, Wiena, February 2001
17. T. Cinkler, D. Marx, C.P. Larsen, D. Fogaras: *“Heuristic Algorithms for Joint Configuration of the Optical and Electrical Layer in Multi-Hop Wavelength Routing Networks”*, *IEEE INFOCOM 2000*, pp.1000-1009, Tel Aviv, March 2000
18. Cs. Gáspár, G. Makács, T. Cinkler, J. Tapolcai: *”Wavelength Routing with Grooming and Protection”*, IFIP ONDM 2003, Optical Network Design and Modelling, Budapest, Hungary, February 2003
19. S. Johansson: *“Transport Network Involving a Reconfigurable WDM Network Layer - A European Demonstration”*, *IEEE Journal on Lightwave Technology*, vol.14, no.6, pp. 1341-1348, June 1996
20. COST 266: http://www.ure.cas.cz/dpt240/cost266/index.html; *“COST 266 Reference Scenario”*, http://ibcn.atlantis.rug.ac.be/projects/COST266_IST_lion/NRS/index.html, January 2002

Managing Traffic Demand Uncertainty in Replica Server Placement with Robust Optimization

Kin-Hon Ho, Stylianos Georgoulas, Mina Amin, and George Pavlou

Centre for Communication Systems Research, University of Surrey, GU2 7XH, UK
{K.Ho, S.Georgoulas, M.Amin, G.Pavlou}@surrey.ac.uk

Abstract. The replica server placement problem determines the optimal location where replicated servers should be placed in content distribution networks, in order to optimize network performance. The estimated traffic demand is fundamental input to this problem and its accuracy is essential for the target performance to be achieved. However, deriving accurate traffic demands is far from trivial and uncertainty makes the target performance hard to predict. We argue that it is often inappropriate to optimize the performance for only a particular set of traffic demands that is assumed accurate. In this paper, we propose a scenario-based robust optimization approach to address the replica server placement problem under traffic demand uncertainty. The objective is to minimize the total distribution cost across a variety of traffic demand scenarios while minimizing the performance deviation from the optimal solution. Empirical results demonstrate that robust optimization for replica server placement can achieve good performance under all the traffic demand scenarios while non-robust approaches perform significantly worse. This approach allows content distribution providers to provision better and predictable quality of service for their customers by reducing the impact of inaccuracy in traffic demand estimation on the replica server placement optimization.

1 Introduction

Content Distribution Networks (CDNs) aim to efficiently deliver web content from servers to users through the Internet. In order to achieve this goal, CDN providers replicate their server infrastructure in multiple locations. The replication technique brings two major advantages to CDNs: first, it minimizes content download time since the replicated servers can be placed quite close to the requesting users; and, second, it allows the CDN providers to operate seamlessly if one of its servers is not available. For simplicity, we call each replicated server a *replica* in this paper.

To efficiently deliver web contents, a CDN provider needs to decide where to place replicas and how many replicas are required, so as to optimize network performance [1,2] and to support Quality of Service (QoS) guarantees [3,4]. This is known as the Replica Server Placement (RSP) problem. Achieving optimal and predictable RSP design is extremely important for the success of the CDN business as users will abandon a web site that fails to provide content in an acceptable response time[1]. In the

[1] According to Zona Research [19], about $4.35 billion may be lost in online business revenues in 1999 due to unacceptably slow response times.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 727 – 739, 2006.

literature, the RSP problem has been formulated as the *minimum P-median problem*, taking as input a set of estimated traffic demands. In theory, if the traffic demands are perfectly known, then an optimal and predictable performance for the RSP can be obtained. Unfortunately, deriving accurate traffic demands is far from trivial since Internet traffic patterns change over time as a result of unpredictable user behavior in traffic request. In addition, perfect (noiseless) flow measurements are rarely available on all links and egress/ingress points of the network [5]. Hence, traffic demands are usually derived with a degree of uncertainty whose consequence is to prevent conventional RSP optimization from producing optimal and predictable performance; this may subsequently lead to potential loss in business revenues. In this paper, we argue that it is insufficient to optimize CDNs performance for only a particular set of traffic demands given relevant inaccuracy. Instead, we need to fundamentally rethink the way in which we design CDNs for coping with uncertainty so as to avoid 'risky' solutions characterized by unsatisfactorily high traffic demand uncertainty in order to sustain, at least, stable business revenues. To the best of our knowledge, this important issue has not been investigated in the literature.

Rather than assuming accurate traffic demand estimation, which is not possible as explained, we propose an approach based on the principles of ***Scenario-based Robust Optimization*** (***SRO)*** [6,7]. When applied to the RSP, SRO structures uncertainty by using a set of potential traffic demand *scenarios*, each exhibits structural difference in traffic volume distribution. The objective of this robust RSP is thus to optimize performance objectives across a variety of such scenarios. We formulate the robust RSP problem as an integer programming problem and solve it by the MINLP solver. Simulation results demonstrate that the robust RSP approach achieves significantly better performance than non-robust optimization approaches under traffic demand uncertainty. In addition, the robust RSP approach guarantees the resulting performance to be within a specified envelope from the optimal solution.

The rest of this paper is organized as follows. In Section 2, we review the deterministic RSP problem formulation. We then present a robust version of RSP in Section 3. Section 4 presents three alternative strategies to tackle the robust RSP problem, which are used for performance comparison. In Sections 5 and 6, we present our evaluation methodology and simulation results. Finally, we conclude the paper in Section 7.

2 Deterministic Replica Server Placement Problem

Table 1 shows the notation used throughout this paper. The deterministic (i.e. non-robust) version of RSP [1] can be summarized as follows. A CDN network is modeled as a graph *G=(N,E)* where N and E are network nodes and links. Given a set of user nodes $I\subseteq N$ and a set of potential server nodes $J\subseteq N$, select P out of J to be server nodes[2] and assign each user traffic demand to the closest server. A distribution cost $d_i c_{i,j}$ is incurred if traffic demand d_i is assigned to server node j, where $c_{i,j}$ is a general cost that may represent hop count, IGP cost, delay or any other performance metric.

[2] In line with [1,2], we assume a full replication for content distribution, i.e. each server has large storage capacities to hold the whole contents for serving any user request.

Table 1. Notation

NOTATION	DESCRIPTION
I	A set of user nodes, indexed by i
J	A set of potential server nodes, indexed by j
S	A set of traffic demand scenarios, indexed by s. This includes the base and the developed traffic demands
d_i	Traffic demand from user node i
$d_{s,i}$	Traffic demand of user node i under scenario s
$c_{i,j}$	Cost to transport one unit from server node j to user node i
X_j	A variable indicating whether node j is selected as server node
$Y_{i,j}$	A variable indicating whether traffic demand of user node i is assigned to server node j
z_s^*	The optimal total distribution cost under scenario s if input data is perfectly known for that scenario

Since the most concerned performance metric for RSP is content download time [8], we assume $c_{i,j}$, to be the delay between i and j over the shortest path in terms of hop count. The goal of RSP is thus to select P nodes as server nodes so as to minimize thetotal distribution cost[33] over all traffic demands. In [1], the RSP problem has been proven NP-hard by mapping it to the *uncapacitated minimum P-median problem*. The problem formulation of the deterministic RSP can be summarized as follows:

$$Minimize \sum_{i\in I}\sum_{j\in J} d_i c_{i,j} Y_{i,j} \tag{1}$$

subject to the following constraints:

$$\forall i \in I : \sum_{j\in J} Y_{i,j} = 1 \tag{2}$$

$$\forall i \in I, j \in J : Y_{i,j} \le X_j \tag{3}$$

$$\sum_{j\in J} X_j = P \tag{4}$$

$$\forall i \in I, j \in J : X_i, Y_{i,j} \in \{0,1\} \tag{5}$$

Objective function (1) minimizes the total distribution cost over all traffic demands. Constraint (2) ensures that each traffic demand is assigned to one server. Constraint (3) ensures that, whenever traffic demand d_i is assigned to node $j\in J$, then j must have been selected as server node. Constraint (4) states that P out of N servers are to be selected. Constraint (5) is the standard integrality constraint.

3 Robust Replica Server Placement Problem

In this section, we present a Scenario-based Robust Optimization (SRO) approach for RSP optimization to manage traffic demand uncertainty. SRO is a comprehensive

[3] Since the distribution cost takes into account the link delay, minimizing the total distribution cost over all traffic demands is effectively equivalent to minimizing the content download time for these traffic demands.

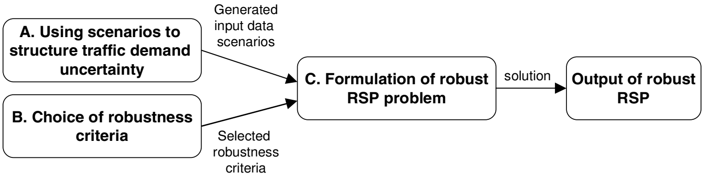

A. Using scenarios to structure traffic demand uncertainty
Generated input data scenarios
B. Choice of robustness criteria
Selected robustness criteria
C. Formulation of robust RSP problem
solution
Output of robust RSP

Fig. 1. Scenario-based robust optimization framework for the replica server placement problem

mathematical programming framework for robust decision making. The SRO framework applied to the RSP problem consists of three elements, as depicted on Figure 1.

A. Using Scenarios to Structure Traffic Demand Uncertainty

Decision makers may develop discrete scenarios that provide visions of alternative possible futures, and then use these them to structure their uncertain input data. Thus, scenarios are devised for representing the decision maker's perceptions about alternative environments in which their decisions might be applied, in the most appropriate manner based on internal knowledge and experience.

When applied to the RSP, SRO models uncertainty as a set of potential traffic demand scenarios. This set of traffic demand scenarios cover at least different critical views of possible traffic characteristics instant, e.g. morning, afternoon and evening. In fact, since the sources of errors or fluctuations in the traffic demands are well understood, their magnitude can be estimated within some known accuracy [5,21].

B. Choice of Robustness Criteria

Recall that the aim of the scenario-based robust optimization is to produce decisions that will have a reasonable objective value under any potential input data scenario. Different criteria can be used to select among robust decisions. We apply two criteria [6] to our robust RSP optimization, which make it more suitable for CDN providers to consider from a practical point of view.

The first criterion is ***minimax***, which aims at getting the best out of the worst possible conditions. This criterion is chosen based on a general observation that the decision makers are (in many cases) risk-averse, meaning that the RSP solution CDN providers want is neither the "optimal" for a particular traffic demand scenario nor the "worst" for any scenario but one that performs reasonably well across all the scenarios. Such risk-averse behavior may also be observed from capacity overprovisioning employed by top-tier Internet service providers as a means to provide good service to all IP traffic in their backbone networks [18]. Hence, CDN providers may want to optimize the worst-case network performance in order to prevent severe unpredicted performance degradation and the need for future expensive network capacity upgrading. The minimax criterion can thus be expressed by minimizing the worst-case total distribution cost across the set of traffic demand scenarios, i.e.

$$Minimize \ \underset{\forall s \in S}{Max}\, F(s) \tag{6}$$

where *F(s)* is the resulting total distribution cost under traffic demand scenario *s*.

Although the minimax criterion can produce reasonably good performance across all the traffic demand scenarios, it may lead to RSP solutions that are overly conservative

Table 2. Example of total distribution cost under four different traffic demand scenarios

Scenario Solution	s_1	s_2	s_3	s_4
x_1	89	90	87	**93**
x_2	79	81	75	**95**
optimal	74	76	70	82

or pessimistic, thereby accepting unnecessary high costs in non-worst-case scenarios. We illustrate this conservative effect by the example in Table 2.

Two solutions, x_1 and x_2, produce different total distribution costs for four traffic demand scenarios (s_1, s_2, s_3, s_4). The solution named "optimal" represents the optimal total distribution cost for each scenario if the traffic demands of that scenario are perfectly known. If only the minimax criterion of equation (6) is considered, x_1 is the best solution since it has lower worst-case total distribution cost than that of x_2 (93 vs. 95). However, x_1 has higher cost than x_2 under scenarios s_1, s_2 and s_3, and their costs deviate highly from the optimal ones. One may observe that if s_1, s_2 or s_3 occurs, x_1 will no longer be the best solution except only for the case where s_4 occurs. However, the occurrence probability (*prob*) of s_4 is likely going to be less than that of s_1, s_2 and s_3 altogether, i.e. $prob(s_4) < prob(s_1) + prob(s_2) + prob(s_3)$.

Ideally, CDN providers may want to obtain a RSP solution that not only has good worst-case total distribution cost but also has the total distribution cost as close as possible to the optimal in each scenario. We therefore employ as the second criterion the minimization of ***relative regret***. The relative regret of a solution in a given scenario is defined as the performance difference in percentage between the solution in that scenario and the optimal solution for that scenario. Thus, CDN providers may seek a RSP solution that keeps the worst-case total distribution cost as low as possible while minimizing the performance deviation of each scenario from optimality.

By jointly optimizing the minimax and relative regret criteria, a bi-criteria robust RSP problem is formulated. The solution that simultaneously optimizes both objectives is called *pareto-optimal*. However, as shown in the example of Table 2, the two objectives may conflict with each other and balancing relevant trade-offs is non-trivial, in particular how to determine their weighted importance. We thus resort to using the $\in$-constraint method [20], which is one of the most favored methods of generating pareto-optimal solutions. In this technique, one objective is selected for optimization, while the other one is constrained so as not to exceed a tolerance value ($\in$). We apply the $\in$-constraint method to the robust RSP by placing a constraint on the relative regret that may be attained by the solution while optimizing the worst-case total distribution cost across all the scenarios. More specifically, the constraint dictates that the relative regret in any scenario must be no greater than $\in$, where $\in \geq 0$. In other words, the cost under each scenario must be within $100(1+\in)\%$ of the optimal cost for that scenario z_s^*. By successively adjusting $\in$, one can obtain solutions with smaller relative regret but greater worst-case total distribution cost and vice versa. One objective of this paper is to demonstrate empirically that substantial improvements in robustness can be attained without large increases in the worst-case total distribution cost.

C. Problem Formulation

By taking into consideration the minimax and the relative regret criteria, we revise the deterministic RSP problem into a robust RSP problem. The optimization objective of the robust RSP problem is to

$$Minimize \ \underset{s\in S}{Max}\sum_{i\in I}\sum_{j\in J} d_{s,i}c_{i,j}Y_{i,j} \tag{7}$$

subject to the following constraints:

$$\forall i \in I : \sum_{j\in J} Y_{i,j} = 1 \tag{8}$$

$$\forall i \in I, j \in J : Y_{i,j} \leq X_j \tag{9}$$

$$\sum_{j\in J} X_j = P \tag{10}$$

$$\forall i \in I, j \in J : X_i, Y_{i,j} \in \{0,1\} \tag{11}$$

$$\forall s \in S : \sum_{i\in I}\sum_{j\in J} d_{s,i}c_{i,j}Y_{i,j} \leq (1+\in)z_s^* \tag{12}$$

Constraints (8)-(11) are identical to constraints (2)-(5). Constraint (12) enforces the $\in$-constraint condition. Compared to the deterministic RSP, which minimizes the total distribution cost for a particular traffic demand scenario, the robust RSP optimizes the worst-case total distribution cost across a variety of traffic demand scenarios, as expressed by the objective function (7). On the other hand, it is not surprising that the robust RSP problem is NP-hard since it is an extension of the deterministic RSP problem, which is itself NP-hard [1]. When the number of traffic demand scenarios $|S| = 1$ and $\in = \infty$, the robust RSP problem reduces to the deterministic one.

4 Alterative Strategies for Managing Traffic Demand Uncertainty

Our implementation of the robust RSP is only one of several methods that can be used to help dimension a network under traffic demand uncertainty. Some common alternative approaches, such as mean-value model, worst-case model and stochastic optimization, can be considered for performance comparison. When applied to the RSP, these approaches differ in their structural traffic volume distribution.

A. Mean-Value Model

In the mean-value model, each element of the *mean traffic demand* is defined as:

$$\bar{d}_i = \sum_{s\in S} d_{s,i} / |S| \qquad \forall i \in I \tag{13}$$

where $|S|$ is number of traffic demand scenarios. The mean traffic demand is then taken as input to solve the deterministic RSP problem (Eq. 1-5).

B. Worst-Case Model

In the worst-case model, each element of the *worst-case traffic demand* is defined as:

$$\hat{d}_i = \underset{s\in S}{Max}\, d_{s,i} \quad \forall i \in I \tag{14}$$

In a similar fashion to the mean-value model, this worst-case traffic demand is then taken as input for solving the deterministic RSP. Note that the total traffic volume of the worst-case traffic demand serves as upper bound of the other traffic demands.

C. Stochastic Optimization (Expected Value Criterion Model)

Stochastic optimization is typically used for solving decision-making problems under risk situations. In the context of stochastic optimization, the expected value criterion model is commonly used. The model seeks the minimization of the expected total distribution cost over all traffic demand scenarios. The input data of the RSP assumes that each traffic demand scenario is probabilistic and the optimization objective is to

$$minimize \sum_{s\in S} \alpha_s \sum_{i\in I} \sum_{j\in J} d_{s,i} c_{i,j} Y_{i,j} \tag{15}$$

where α_s is the occurrence probability of traffic demand scenario *s*. Without loss of generality, we assume in this paper each traffic demand scenario has equal occurrence probability. This assumption is also known as *Laplace criterion* [9] for decision making under uncertainty. The Laplace criterion is based on the *principle of insufficient* reason. It asserts that, if one is completely unaware of which scenario will happen, then these scenarios may be treated as equally likely, since there is no reason to believe otherwise.

5 Evaluation Methodology

A. Network Topology

Our simulation is performed on 30-node AS-level topologies with node degree of 3, generated by the BRITE topology generator [10]. Each node and link in the topology represents an AS and a physical link between ASes respectively. Each link is associated with a propagation delay generated by BRITE. We assume that all ASes are user nodes and they are also considered as potential server nodes. The cost between two ASes (i.e. $c_{i,j}$) is the sum of link propagation delays along the shortest AS-hop path. Since the communication cost within an AS is often much better than between different ASes, we assume that at most one replica can be placed within each AS and we neglect the distribution cost generated by users attached to the same AS as the replica.

B. Web Content Traffic Demand

We generate synthetic traffic demands for our evaluation. We attach a traffic demand to each AS, which represents the total traffic demand requested at the AS. Previous work has shown that web traffic is not uniformly distributed. According to [11], the popularity of web content follows a Zipf-like distribution of $y{\sim}x^{-\alpha}$, which is a widely adopted model for real Web traces. The default value of popularity parameter α is set to be 0.75 with a reference to the analysis of real Web traces in [11].

We generate traffic demand scenarios using the methodology proposed in [6]: the traffic demand can vary within known ranges or can be estimated within known accuracy. This range is denoted by an error margin parameter $\omega \geq 1$. We consider ***base traffic demand*** which can be thought of as our best “guess” of the actual traffic demand. The set of applicable traffic demand scenarios, which we call ***developed traffic demands***, includes each scenario *s* with error margin such that

$$d_{s,i} = \{\xi \in \mathbb{R} : d_i / \omega \le \xi \le \omega d_i\} \qquad \forall i \in I \tag{16}$$

These developed traffic demands can be thought of as corresponding to traffic fluctuation or random errors in traffic estimation. The above method for generating traffic demand scenarios has also been used to evaluate many practical optimization problems [6] such as the robust Knapsack Problem. We remark that this traffic demand generation process is our best attempt to model web traffic fluctuation, as no synthetic model for the actual behavior of traffic in real networks can be found in the literature.

C. Comparison of RSP Approaches

We compare the performance of the following RSP approaches in our simulation:

- **Deterministic:** we run the deterministic RSP individually for each of the base and the developed traffic demands. We then use each of these RSP solutions to obtain the total distribution cost that would be achieved by the other traffic demand scenarios. In our simulation, we denote as "base" the deterministic optimization taking the base traffic demand as input. Likewise, the term "first" denotes the deterministic optimization taking the first of the developed traffic demands as input and so forth.
- **Statistical:** we run the mean-value and the worst-case models. These models roduce their traffic demands using the base and the developed traffic demands. We denote as "mean" and "worst" the two models respectively.
- **Robust:** we run the robust RSP approach by taking the base and the developed traffic demands as input data scenarios. We denote this approach as "robust".
- **Stochastic:** we run the stochastic optimization (i.e. the expected value criterion model) by taking the base and the developed traffic demands as input data scenarios. We denote the stochastic optimization as "stochastic".

D. Performance Metrics

The following two performance metrics [7] are used to evaluate different RSP approaches. For these metrics, lower values are better than high values.

- **Solution robustness:** an RSP solution is robust to the total distribution cost if it performs reasonably well for any realization of the traffic demand scenarios $s \in S$. For this metric, we capture the worst-case (i.e. the highest) total distribution cost under all the traffic demand scenarios for each RSP approach.
- **Relative robust deviation:** we capture the maximum relative regret under all the traffic demand scenarios for each RSP approach.

6 Simulation Results

All the RSP approaches presented in this paper have been implemented using the AMPL modeling language [12] and solved by the Mixed Integer Nonlinear Programming

(MINLP) solver [13][4]. The MINLP solver implements a branch and bound algorithm searching a tree whose nodes correspond to continuous nonlinear optimization problems. The continuous problems are solved using filterSQP, a Sequential Quadratic Programming solver, which is suitable for solving large nonlinear problems.

An important element in our simulation is the generation of various traffic demand scenarios. Following the methodology described in Section 5-B, we generate a base traffic demand and five developed traffic demands. Each simulation result takes approximately 10 minutes running time on average.

A. Evaluation of Solution Robustness

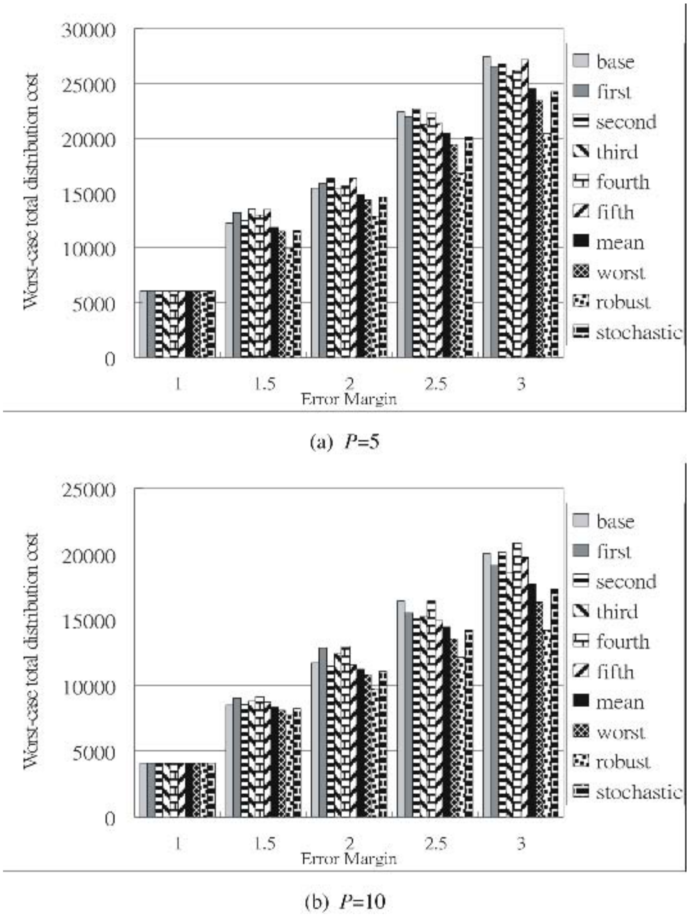
In this section, we evaluate the solution robustness of different RSP approaches. Regarding the value of $\in$, we initially set it to ∞ and then evaluate its impact on the worst-case total distribution cost and relative regret in the subsequent sections.

Figures 2(a) & (b) show the worst-case total distribution cost as a function of error margin for P=5 and P=10 respectively, where P is the number of servers to be selected. Similar result patterns for all the RSP approaches are exhibited in the two figures. An obvious difference between them is that the higher the P, the lower the worst-case total distribution cost because more servers can be located closer to the users. Therefore, we make a performance analysis that is applicable to both P results.

When ω=1.0, all the RSP approaches produce identical performance because they use identical traffic demand. At all other values of error margin, we observe a general phenomenon that the deterministic approach (“base”, “first”…“fifth”) is the worst performer. This result is expected: in fact the RSP solution optimized for a particular-traffic demand scenario may no longer maintain optimality for the other scenarios that have different structural traffic distribution. The performance gets worse when the error margin is large. In contrast, the statistical approach (i.e. “mean” and “worst”) has slightly better performance than the deterministic approach. For the mean-value model, since the mean traffic demand is mixture of traffic characteristics from different traffic demand scenarios, it usually performs better than the deterministic approach that optimizes for only one traffic demand scenario. On the other hand, the worst-case model performs even slightly better than the mean-value model since the worst-case performance is optimized; this is close to the optimization objective of the robust RSP. This model, however, is overly conservative and does not produce truly optimal performance under traffic demand uncertainty. This is demonstrated by the superior performance of the robust approach.

Unlike the others, the worst-case total distribution cost of the robust approach increases smoothly as the error margin increases. This shows that the performance is not overly sensitive to errors in traffic demand estimation. Compared to the robust approach, the stochastic approach can only perform as well as the mean-value model. This phenomenon is expected because both approaches would behave optimally in the mean. However, they show poor performance at some particular realization of

[4] Ideally, heuristics are proposed to handle large-scale NP-hard optimization problems. However, since this paper aims at demonstrating the effectiveness of the robust RSP approach on coping with traffic demand uncertainty, we therefore solve the RSP problem using mathematical programming. Nevertheless, we are motivated to devise efficient heuristics to solve the problem as our future work.



(a) P=5

(b) P=10

Fig. 2. Worst-case total distribution cost vs. error margin

scenarios. On the whole, for coping with traffic demand uncertainty, the robust approach has significantly minimized the worst-case total distribution cost over the deterministic approach (about 15%-30% across different error margin values) and both the statistical and stochastic approaches (about 8%-20%).

B. Evaluation of Relative Robust Deviation

We would like to know for the results presented so far how much their performance deviates from the optimal one. We present the relative robust deviation results in Table 3. For brevity, we only show the results for P=10 and error margin equal to 2.0. For all other values of error margin, we have reached a similar conclusion.

The results in Table 3 can be interpreted as follows. Each row represents the solution of a given RSP approach, and each column represents a traffic demand scenario. The value of the table element α_{ij} (that is row i, column j) corresponds to the relative regret that would result for traffic demand scenario j if the solution of RSP approach i was implemented. Therefore, the values on the diagonal represent zero relative regret.

The maximum relative regret under all the traffic demand scenarios for each RSP solution in the row is shown in bold and underline face. The results show that the deterministic approach has the highest maximum relative regret. The robust approach is the best performer followed by the statistical and the stochastic approaches. This result is similar to that in Figure 2(b); in general, the higher the worst-case total distribution cost, the higher the maximum relative regret.

[5] **Table 3.** Relative regret (in %) for P=10 and ω= 2.0

Scenario Solution	*base*	*first*	*second*	*third*	*fourth*	*fifth*	*mean*
base	0	17.23	18.23	23.04	**29.43**	17.54	15.21
first	22.24	0	**35.11**	25.32	26.51	19.24	14.31
second	21.45	19.92	0	27.34	24.12	**28.72**	17.9
third	22.45	15.33	19.37	0	**30.18**	14.35	12.45
fourth	19.26	27.21	28.23	19.62	0	**38.12**	20.59
fifth	**27.78**	23.57	25.12	19.43	27.27	0	18.56
mean	12.45	14.39	14.63	**20.21**	18.41	12.77	0
worst	10.22	16.32	9.56	14.56	**20.45**	14.13	7.72
robust	7.11	**11.67**	5.12	10.67	5.22	4.25	3.24
stochastic	14.15	12.21	14.13	16.24	**21.31**	13.74	9.43

C. Evaluation of $\in$

One of the objectives of the robust RSP is to minimize the maximum relative regret (by the choice of$\in$) with as little increase in the worst-case total distribution cost as possible. To illustrate this trade-off, we used the constraint method of multi-objective programming [14] to generate a trade-off table between the worst-case total distribution cost and the maximum relative regret. In particular, we first solved the problem with $\in = \infty$ and recorded the two performance metrics; we then set $\in$ equal to the maximum relative regret minus a small step down value (e.g. 0.2%) and re-solved the problem, continuing this process until no feasible solution could be found for a given value of $\in$. We performed this experiment using the traffic demand scenarios with error margin equal to 2.0 and P=10. The results are summarized in Table 4.

Table 4. Worst-case total distribution cost versus maximum relative regret

$\in$	*Total Distribution Cost*	*% Increase*	*Max Rel Reg*	*% Decrease*
∞	9753	0.0%	11.67%	0.0%
0.1147	9806	0.55%	11.27%	3.42%
0.1107	9863	1.13%	9.89%	15.25%
0.0969	10102	3.57%	8.54%	26.82%
0.0834	10187	4.46%	7.43%	36.33%

The column marked "$\in$" gives the value of $\in$ used to solve the problem; "*Total Distribution Cost*" is the worst-case total distribution cost; "*% Increase*" is the percentage by which the worst-case total distribution cost is greater than that of the found

[5] Traffic demand produced from the worst case model is not included in the table column since it has higher traffic volume than the other traffic demand scenarios.

solution using $\in = \infty$; "*Max Rel Reg*" is the maximum relative regret of the best found solution; and "*% Decrease*" is the percentage by which the maximum relative regret is smaller than that of the found solution using $\in = \infty$. It is clear that large reductions in the maximum relative regret are possible with only small increases in the worst-case total distribution cost. For example, the last solution represents a 36.33% reduction in the maximum relative regret with only less than a 4.46% increase in the worst-case total distribution cost. These results justify the use of the $\in$-constraint approach since it costs very little to achieve robustness.

D. Performance Summary of the RSP Approaches

The simulation study in this section evaluated the performance of different RSP approaches. Simulation results have shown that the robust approach produces significantly better worst-case total distribution cost than non-robust approaches under traffic demand uncertainty. The robust approach also guarantees the performance of the solution to be within a specified envelope from the optimal solution, thereby improving robustness on RSP performance. We therefore conclude that the robust RSP approach can make RSP performance more robust and predictable.

7 Conclusions

In this paper we faced the problem of RSP, assuming that traffic demand uncertainty is handled by a set of traffic demand scenarios. By using the principles of scenario-based robust optimization, we proposed a novel integer programming formulation for robust RSP. We provided empirical results to assess the performance of several commonly used techniques for robust RSP. The results show that the robust RSP approach, whose optimization runs across the set of traffic demand scenarios, significantly improves the solution robustness while it also minimizes the performance deviation from the optimal solutions. We believe that our work provides insights to CDN providers on how to design robust CDNs by reducing the impact of inaccuracy in traffic demand estimation so as to provision better and predictable QoS for their users and avoid potential loss in business revenues.

We emphasize that our idea of SRO is not only limited to the RSP problem. In fact, the CDN-related design problems to which it can be applicable are *numerous*. Examples are web object replication [15,18], request routing [18], cache location [16] and topological design for service overlay networks [17]. A common characteristic of these CDN design problems is that their optimization objectives are influenced by the accuracy of estimated traffic demands. We believe that the robust approach can be adopted by CDN providers as a means to make their networks more robust.

Acknowledgement

This work was undertaken in the context of FP6 Information Society Technologies AGAVE (IST-027609) project, which is partially funded by the Commission of the European Union.

References

1. L. Qiu et al., “On the Placement of Web Server Replicas,” Proc. *IEEE INFOCOM*, 2001.
2. S. Jamin et al., “Constrained Mirror Placement on the Internet,” Proc. *IEEE INFOCOM*, 2001.
3. X. Tang and J. Xu, “QoS-Aware Replica Placement for Content Distribution,” *IEEE Transactions on Parallel and Distribution Systems*, 16(10), 2005, pp. 921-932.
4. G. Rodolakis et al., “Replicaed Server Placement with QoS Constraints,” Proc. *3rd International Workshop on QoS in Multiservice IP Networks (QoSIP)*, 2005.
5. A. Feldmann et al.., “Deriving Traffic Demands for Operational IP Networks: Methodology and Experience,” *IEEE/ACM Transactions on Networking*, 9(3), 2001, pp. 265-280.
6. P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*, Kluwer Academic Publishers, 1997.
7. J.M. Mulvey et al., “Robust optimization of large-scale systems,” *Operations Research*, 43, 1995, pp. 264-281.
8. N. Hu et al., “Optimizing Network Performance in Replicated Hosting,” Proc. *IEEE International Workshop on Web Caching and Content Distribution (WCW)*, 2005.
9. H.A. Taha. Operations Research, 7th edition, Prenticall Hall, 2003.
10. A. Medina et al., “BRITE: An Approach to Universal Topology Generation,” Proc. *MASCOTS 2001*, 2001.
11. L. Breslau et al., “Web Caching and Zipf-like Distributions: Evidence and Implications,” Proc. *IEEE INFOCOM*, 1999.
12. A Modeling Language for Mathematical Programming. Available at www.ampl.com.
13. The MINLP solver. University of Dundee, UK.
14. J.L. Cohon. *Multiobjective programming and planning*. Mathematics in Science and Engineering. Academic Press, New York, 1978.
15. J. Kangasharju et al., “Object replication strategies in content distribution networks,” *Computer Communications*, 25(4), 2002, pp. 376-383.
16. P. Krishnan et al., “The cache location problem,” *IEEE/ACM Transactions on Networking*, 8(5), 2000, pp.568-582.
17. S.L. Vieira and J. Liebeherr, “Topology design for service overlay networks with bandwidthi guarantees,” Proc. *IEEE IWQOS*, 2004, pp. 211-220.
18. T. Bektas et al., “Designing cost-effective content distribution networks,” to appear in *Computers & Operations Research*, 2006.
19. The economic impacts of unacceptable web-site download speeds. Zona Research, 1999.
20. V. Chankong and Y.V. Haimes. *Multiobjectve Decision Making–Theory and Methodology*, Elsevier, New York, 1983.
21. Y. Zhang et al., “An Information-Theoretic Approach to Traffic Matrix Estimation,” Proc. *ACM SIGCOMM*, 2003.

An Information Theoretic Approach for Systems with Parallel Distributions: Case Studying Internet Traffic

Charalabos Skianis[1],[2] and Lambros Sarakis[2]

[1] University of the Aegean,
Department of Information and Communication Systems Engineering,
GR-83200, Karlovassi, Greece
[2] National Centre for Scientific Research 'Demokritos',
Institute of Informatics & Telecommunications,
15310 Aghia Paraskevi Attikis, POB 60228, Athens, Greece
{skianis, sarakis}@iit.demokritos.gr

Abstract. The principle of Minimum Relative Entropy (MRE) is applied to characterize a 'proportionality' relationship between the state probabilities of infinite and finite capacity queues at equilibrium and thus, establish an information theoretic interpretation for the exact global balance solution of some finite capacity queues with or without correlated arrival processes. This result serves to establish the utility of the MRE inference technique and encourage its applicability to the analysis of more complex, and thus more realistic, queuing systems. The principles of Maximum Entropy (ME) and MRE are then employed, as least-biased methods of inference, towards the analysis of a Internet link carrying realistic TCP traffic, that exhibit this 'proportionality' relationship between a finite and infinite buffer system, as produced by a large number of connections. The analytic approximations are validated against exhaustive simulation experiments. Despite its simplicity, the methodology captures the behavior of the system under study both in the cases of finite and infinite buffers and finally and can easily be utilized for network management and design, capacity planning, and congestion control.

1 Introduction

Nowadays Internet traffic is carried by networks using primarily TCP as the transport protocol. Research efforts in the field span from direct measurements and TCP protocol simulations to analytical modeling towards more elaborate problems such as network management and design, capacity planning and congestion control. In this paper, we present a simple information theoretic approach to the case of both uncongested and congested links. The specific topic is extensively studied with many efforts devoted to study characteristics of Internet traffic such as long range dependence, self-similarity, and multi-fractal scaling (e.g. [1], [2], [3]). In certain cases, traditional approaches proved inadequate to handle the complexity of the packet arrival process with new theoretical tools suggested in [4] for traffic engineering purposes. Packet-level characteristics of Internet traffic is studied in [5], [6], [7], [8] and finite size

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 740 – 751, 2006.

TCP connections studied in [9], [10], [11]. In [12] the notion of batch arrivals is used for the study of link fed with a varying (large) number of finite TCP flows. The approach although close to simulation results under a set of assumptions proves somehow elaborate and time consuming with the calculation of the batch size distribution and the use of iterative processes in the finite buffer cases. In a similar fashion [13] considers short TCP flows that never leave slow-start and uses a burst-level M/G/1 model to evaluate the queue length distribution at the bottleneck router for the infinite buffer case. To our knowledge there is no simple and robust suggestion towards the estimation of the queue length distribution of an Internet link (with finite or infinite buffer) carrying realistic traffic as produced by converging TCP flows.

Classical queuing theory provides a conventional and powerful framework for formulating and solving models of discrete flow systems such computer systems, communication networks and flexible manufacturing systems. However, in many cases, simplified assumptions are employed in order to produce tractable solutions, whilst approximate methods are required to analyze more complex, and thus, more realistic models. Since mid-60s, alternative ideas and tools have been proposed in the literature ([14], [15], [16]).

It can be argued that one of the most fundamental requirements for the analysis of complex queuing systems is the provision of a convincing interpretation for a probability assignment free from arbitrary assumptions. In a more general context, this was the motivation behind the principles of Maximum Entropy (ME) ([17], [18]) and Minimum Relative Entropy (MRE) ([19], [20]). These principles provide self-consistent methods of inference for estimating uniquely and in a least biased fashion an unknown but true probability distribution, based on information expressed in terms of known to exist prior distribution estimate and/or true mean value constraints. Over the recent years, these principles have been applied to characterize useful information theoretic approximations of performance distributions for queuing systems and networks and the formation of experimental performance bounds (e.g., [21], [22], [23]).

In this paper, the principle of MRE is used to determine a 'proportionality' relationship between the state probabilities of some infinite and finite capacity queues and thus, establish an information theoretic interpretation of global balance solutions for finite capacity queues with or without correlated arrival processes. The MRE methodology can be further applied to characterize exact and approximate relationships for more complex queuing systems for which the 'proportionality' relationship may not have as yet been established via classical queuing theory. Furthermore, the MRE principle may lead to cost-effective closed form solutions and performance bounds for finite capacity queues which in turn can be used as cost-effective building blocks for the performance analysis of Queueing Network Models (QNMs) of discrete-flow systems. An example of such application can be seen in [23].

The main contribution of this paper can be summarized as follows: (i) the principle of MRE is used to determine a 'proportionality' relationship between the state probabilities of some infinite and finite capacity queues and thus, establish an information theoretic interpretation of global balance solutions for finite capacity queues; (ii) we focus on an Internet link with traffic resulting out of converging flows of TCP connections and show that a ‘proportionality’ relationship exists between finite and infinite capacity systems; (iii) we introduce the information theoretic concepts of ME and MRE, as least-biased methods of inference, towards the estimation of the queue length distribution, for both finite and infinite buffer systems, based on a small set of system characteristics; (iv) we show the good fit of our approach.

The principles of ME and MRE, a generalization, are described in Section 2 whilst Sections 3 and 4 present the analysis of infinite and finite systems based on the ME and MRE principles and discuss on the 'proportionality' relationship amongst finite and infinite systems. Section 5 outlines the system setting and presents the analysis of the infinite and finite systems based on the ME and MRE principles, respectively. Extensive simulation study with ns-2 proves the good fit of the approach and the 'proportionality' relationship between internet links with finite and infinite capacity buffers. Section 6 concludes the work and suggests future directions.

2 The Principles of ME and MRE

Let x be the state of a system with a set D of feasible states. Let D^* be the set of all the probability density functions (pdf) p on D such that $p(x)\geq 0$, $\forall x\in D$, and

$$\int_D p(x)dx = 1 \tag{1}$$

Suppose that the system under consideration is described by a true but unknown density function $p^*\in D^*$ and that $q\in D^*$ is a prior density that is a current estimate of p^*, such that $q(x)\geq 0$, $\forall x\in D$.

In addition, new information for the system places a number of constraints on p^*, in the form of expectations defined on a set of k suitable functions $\{a_i(x)\}$, $i=1,2,\ldots,k$ with known values $\{<a_i>\}$, $i=1,2,\ldots,k$ namely

$$\int_D a_i(x)p(x)dx = < a_i >,\ i=1,2,\ldots,k, \tag{2}$$

where k is less than the number of possible states.

Since the above set of constraints (1)-(2), denoted by $I=(p^*\in\Phi)$ do not determine the form of $p^*(x)$ completely, they are satisfied by a set of pdfs $\Phi\subseteq D^*$.

The principle of MRE (e.g., [20]) states that of all pdfs that satisfy constraints I, the least biased one is the posterior pdf, $p\in\Phi$ that minimizes the relative entropy function, $H(p,q)$ in the set Φ, namely

$$H(p,q) = \min_{p'\in\Phi} H(p',q) \tag{3}$$

where

$$H(p',q) = \int_D p'(x)\log\left({p'(x)}\Big/{q(x)} \right) dx\ . \tag{4}$$

By applying the Lagrange's method of undetermined multipliers the form of the posterior pdf is [20]

$$p(x) = q(x)\exp\left(-\beta_0 - \sum_{i=1}^{k} \beta_0 a_i(x)\right), \tag{5}$$

where $\{\beta_0\}$ and $\{\beta_i\}$, $i=1,2,\ldots,k$ are the Lagrangian multipliers whose values are determined by the constraints (1) and (2).

From (1) the normalizing constant is given by

$$\exp(\beta_0) = \int_D q(x)\exp\left(-\sum_{i=1}^{k} \beta_i a_i(x)\right) dx \ . \tag{6}$$

If the integral in (6) is solved analytically, closed form expressions could be derived for $\{\beta_i\}$, i=1,2,…,k in terms of the mean values $<\alpha_i>$.

In an information theoretic context (c.f.,[17]), the ME solution corresponds to the maximum disorder of system states, and thus is considered to be the least biased distribution estimate of all solutions that satisfy the system's constraints. In sampling terms, Jaynes in [18] has shown that, given the imposed constraints, the ME solution can be experimentally realized in overwhelmingly more ways than any other distribution. Major discrepancies between the ME distribution and the experimentally observed distribution indicate that important physical constraints have been overlooked. Similar justifications can be advanced for relative entropy minimization.

In formal terms, the relative entropy minimization procedure may be seen as an information operator "*o*" that takes two arguments, a prior distribution *q* and a new constraint information *I* of the form (1) and (2), yielding a posterior MRE distribution *p*, i.e. *p=qoI.* To this end it can be shown that minimization of H(*p*,*q*) uniquely characterizes distribution *p*, satisfying four consistency inference criteria proposed by Shore and Johnson in [20]. In particular, it has been shown that ME and MRE solutions are uniquely correct distributions and that any other functional used to implement operator "*o*" will produce the same distribution as the entropy and relative entropy functionals, otherwise it will be in conflict with the consistency criteria.

In the field of systems modeling, expected values of various performance distributions of interest, such as the number of jobs in each resource queue concerned, are often known, or may be explicitly derived, in terms of moments of interarrival and service time distributions. Note that the determination of the distributions themselves, via classical queuing theory, may prove an unfeasible task even for systems of queues with moderate complexity. However, prior estimates of distributions may be obtained by using properties of the system, as appropriate. Hence, it is implied that the methods of entropy maximization and relative entropy minimization may be applied to characterize useful information theoretic approximations of performance distributions of queuing systems and networks.

3 Maximum Entropy Solution of the $M^{[X]}$/G/1 Queue

The $M^{[x]}$/G/1 queues denote, single server queuing systems at equilibrium with infinite capacity, general (G) service times and exponential (M) batch interarrival times, respectively, where x is an independent random variable representing the number of customers within an arriving batch. Let at any given time, *n*, *n*=0,1,…, be the number of packets in the system and $\{q(n)\}$be the true but unknown steady state probability of having *n* packets. Suppose all that is known about the state probabilities *q*(*n*), *n*=0,1,…, is the following set of mean value constraints:

(i) Normalization $\sum_{n=0}^{\infty} q(n) = 1$, (7)

(ii) Probability of a busy system $\sum_{n=0}^{\infty} h(n)q(n) = 1 - q(0)$, (8)

where *h*(*n*)=0, *n*=0, or 1, otherwise,

(iii) Average queue length, <*n*>, $\sum_{n=0}^{\infty} nq(n) = < n >$. (9)

The probability distribution *q*(*n*) can be completely specified by maximizing the entropy functions

$$H(q(n)) = -\sum_{n=0}^{\infty} q(n)\log(q(n) \tag{10}$$

subject to prior information expressed in the form of constraints (7)-(9). By applying the method of Lagrange's undetermined multipliers the ME solution is expressed as

$$q(n) = (1/Z)g^{h(n)}x^n, n = 0,1,..., \tag{11}$$

where *Z*, *g*, *x*, can be determined form the set of constraints (7)-(9).

To this end, the Maximum Entropy solution for *q*(*n*) is given by

$$q(n) = \begin{cases} 1/Z, & n = 0, \\ (1/Z)\, gx^n, & n > 0, \end{cases} \tag{12}$$

where

$$1/Z = q(0),\ x = 1 - (1 - q(0))/ < n >,\ g = (1 - q(0))(1 - x)/(q(0) \cdot x) \tag{13}$$

4 Minimum Relative Entropy Solution of the M$^{[X]}$/G/1/N Queue

The M$^{[x]}$/G/1/N queue denote, single server queuing systems at equilibrium with finite capacity, general (G) service times and exponential (M) batch interarrival times, respectively, where x is an independent random variable representing the number of customers within an arriving batch. It is assumed that all those customers that upon arrival find the buffer full are turned away. At any given time, let n, n=0,1,...,N, be a system state representing the number of customers present in the system and {λ, Ca^2} and {μ, Cs^2} the mean rate and square coefficient of variation (SCV) of the interarrival and service time distributions, respectively. Suppose that $p_N(n)$, n=0,1,...,N, is the true but unknown probability distribution that there are n customers in the system and q(n) is a prior estimate of $p_N(n)$. Moreover, new information takes the form of the following constraints,

(i) The Normalization, $\sum_{n=0}^{N} p_N(n) = 1$, (14)

(ii) The Full Buffer State Probability, $p_N(N) = \varphi,\ 0 < \varphi < 1$, written as (15)

$$\sum_{n=0}^{N} f(n)p_N(n) = \varphi,\ f(n) = \begin{cases} 0, & n < N \\ 1, & n = N \end{cases},$$

and satisfying the flow balance condition

$$\lambda(1 - \pi) = \mu U\ , \tag{16}$$

where, π is the probability that an individual customer will be blocked upon arrival, and U is the utilization.

The form of the MRE solution $p_N(n)$, n=0,1,..., N can be characterized by minimizing the relative entropy function, *H*(*p*$_N$,*q*), subject to constraints (i) and (ii). This can be achieved by applying the method of Lagrange's undetermined multipliers, leading to the MRE solution

$$p_N(n) = (1/Z)q(n)y^{f(n)}, \tag{17}$$

where Z is the normalizing constant and y is the Lagrangian coefficient corresponding to constraint (ii).

4.1 The Proportionality Relationship

For the queuing systems under consideration the stationary distribution, $p_N(n)$, satisfies the following set of equations

$$p_N(n) = p_N(0)\alpha(n) + \sum_{j=1}^{n+1} p_N(j)\alpha(n-j+1), 0 \le n \le N-2, \tag{18}$$

where $\alpha(n)$ is the probability that n messages arrive during a service period. Note that eqs. (18) are of the same form as those for the queue length distribution (qld) $\{p_\infty(n); 0 \le n \le N-2\}$ of the corresponding infinite queuing system, namely

$$p_\infty(n) = p_\infty(0)\alpha(n) + \sum_{j=1}^{n+1} p_\infty(j)\alpha(n-j+1), 0 \le n \le N-2. \tag{19}$$

By dividing the system of eqs. (18) and (19) by $p_N(0)$ and $p_\infty(0)$, respectively, ratios $p_N(n)/p_N(0)$ and $p_\infty(n)/p_\infty(0)$ become identical for $0 \le n \le N-1$. Thus:

$$p_N(n) = (p_N(0)/p_\infty(0))p_\infty(n), \quad 0 \le n \le N-1. \tag{20}$$

4.2 MRE Solution and the Proportionality Relationship

The form of the MRE solution suggests the following proportionality relationships between the unknown probability distribution $p_N(n)$ and the prior estimate for that distribution, *q*(*n*), namely

$$p_N(n) = \begin{cases} (1/Z)q(n), & 0 \le n \le N-1, \\ (1/Z)q(n)y, & n = N. \end{cases} \tag{21}$$

The prior distribution estimate, *q*(*n*) can be determined by the qld of the corresponding infinite capacity M$^{[x]}$/G/1 queues, given in Section 3. The MRE solution for the finite capacity systems, after some mathematical manipulations, can be clearly expressed by proportionality relationship (20) for $0 \le n \le N-1$, whilst

$$p_N(n) = (p_N(0)/p_\infty(0))p_\infty(n)y, \; n = N. \tag{22}$$

The remaining unknown y can be evaluated by using the set of constraints expressed by eqs. (14) and (15) and the flow balance equation (16). Thus, the MRE solution captures the exact solution for the finite capacity system. Moreover, the MRE solution is of closed form and thus it can be used as a cost effective building block towards the analysis of queuing networks with finite capacity.

Two typical case studies involving the Exponential (M) and Generalized Exponential (GE) interevent-time distributions are applying expressions (14)-(16), (20) and (22) are presented below.

4.2.1 M/M/1N Queue

For a stable M/M/1/N queue Lagrangian coefficients Z and y are determined by

$$Z = 1 - \rho^{N+1}, \qquad y = 1. \tag{23}$$

Thus, the MRE solution for the M/M/1/N queue is identical to the exact state probability

$$p_N(n) = (1-\rho)/(1-\rho^{N+1})\rho^N, \quad 0 \le n \le N. \tag{24}$$

4.2.2 GE/GE/1N Queue

The GE type distribution is of the form

$$\Pr(W \le t) = 1 - \tau \exp(-\sigma t), t \ge 0, \tag{25}$$

where $\tau = 2/(1+C^2)$ and $\sigma = \tau\nu$ and W is the random variable of the interevent time, while $1/\nu$ and C^2 are the corresponding mean and SCV.

Moreover, the qld of the infinite capacity GE/GE/1 queue is given by [22]

$$q(n) = p_\infty(n) = \begin{cases} 1-\rho, & n = 0, \\ (1-\rho)gx^n & n = 1,2,..., \end{cases} \tag{26}$$

For a stable GE/GE/1/N queue Z and y, can be determined by using constraints (14)-(15) and the prior distribution of the form (26), namely

$$Z = 1 - \rho^2 x^{N-1}, \; y = (\tau(1-\rho\sigma) + \rho\sigma)/(\tau\sigma), \tag{27}$$

where $\sigma = 2/(1+Ca^2)$ and $\tau = 2/(1+Cs^2)$. Thus, the MRE solution for the GE/GE/1/N queue is given by

$$p_N(n) = \begin{cases} (1/Z)q(n), & 0 \le n \le N-1, \\ (1/Z)q(n)y, & n = N, \end{cases} \tag{28}$$

and turns out identical to the exact solution for the GE/GE/1/N queue. The latter can be derived directly by carrying out a considerable amount of non-trivial algebraic manipulations to solve the system of GE-type global balance equations via the method of Z-transform.

5 Internet Link Carrying Realistic TCP Traffic

The simulation topology used in *ns-2* [24] to validate the described approach is illustrated in Fig. 1. It assumes that there are 100 source and destination nodes connected via a bottleneck link of capacity 50Mbps and two-way propagation delay of 1ms. The capacity of the source and destination links equals that of the bottleneck link. The two-way propagation delays of the source and destination links are uniformly distributed in [2ms,10ms] and [10ms,200ms], respectively. The buffer at the output queue of S/R 1 towards the bottleneck link has a size of *N* packets.

The arrival process of new connections is assumed to be Poisson. When a new connection opens it chooses randomly a source and a destination node and performs a data transfer of *S* packets using the Reno flavor of TCP. The maximum packet size is 1500 bytes and *S* is exponentially distributed with an average of 30 full-size packets. The maximum window size advertised by the receivers is 32 packets.

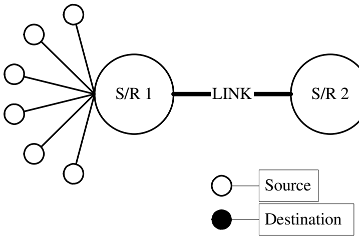


Fig. 1. Internet Link topology

5.1 ME and MRE Solution for the Internet Link

In this section the principles of ME and MRE and classical queuing theory are applied to device a computationally efficient solution for the performance analysis of Internet links for the cases of infinite and finite buffer systems.

5.1.1 ME Solution for the Internet Link with Infinite Buffer

Let at any given time, n, n=0,1,…, be the number of packets in a typical Internet queue as depicted in Fig. 1, $\{q(n)\}$ be the true but unknown steady state probability of having n packets. Suppose all that is known about the state probabilities $q(n)$, n=0,1,…, is the following set of mean value constraints:

(i) Normalization $\sum_{n=0}^{\infty} q(n) = 1$, (29)

(ii) Probability of a busy system $\sum_{n=0}^{\infty} h(n)q(n) = 1 - q(0)$, (30)

where $h(n)$=0, n=0, or 1, otherwise,

(iii) Average queue length, <n>, $\sum_{n=0}^{\infty} nq(n) =< n >$. (31)

Following the reasoning in Section 3, the ME solution for $q(n)$ is given by

$$q(n) = \begin{cases} 1/Z, & n = 0, \\ (1/Z)\, gx^n, & n > 0, \end{cases} \tag{32}$$

where

$$1/Z = q(0),\ x = 1 - (1 - q(0)) / < n >,\ g = (1 - q(0))(1 - x) / (q(0) \cdot x)\,. \tag{33}$$

5.1.2 MRE Solution for the Internet Link with Finite Buffer

Let at any given time n (n=0,1, …, N) be the number of packets in a typical Internet link queue of finite capacity N (>0), $\{p(n)\}$ be the true but unknown steady state probability of having n packets and $q(n)$ be a prior steady state probability estimate of $p(n)$. Moreover new information takes the form of the following mean value constraints:

(i) Normalization, $\sum_{n=0}^{N} p(n) = 1$, (34)

(ii) Full Buffer State Probability, $\{\varphi$=$p(N)$, 0<φ<1$\}$, $\sum_{n=0}^{N} f(n)p(n) = \varphi$, (35)

where $f(n)$=1, if n=N, and 0 otherwise.

In a manner similar to Section 3 the MRE solution for the system is expressed by

$$p(n) = (1/Z)q(n)y^{f(n)}, \quad n = 0,1,...,N, \tag{36}$$

where 1/*Z*, and *y* can be determined by using the set of constraints (34) and (35), and *q*(*n*) a the prior estimate for the pdf.

The prior pdf *q*(*n*) is estimated by the qld of the corresponding infinite queue. This type of prior is a natural choice which enables the MRE approximation pdf *p*(*n*) to capture the exact solution in certain cases (c.f., [23]). In the present analysis the ME solution for the corresponding infinite capacity queue is derived in the previous subsection and proposed as the prior pdf *q*(*n*). The form of the MRE solution suggests the proportionality relationships already discussed in Section 4 between the Infinite and Finite Buffer Systems for the case of Internet Links carrying TCP traffic.

5.1 Validation

The 'proportionality' relationship between infinite and finite buffer cases is exhibited in Fig. 2 via a set of typical numerical experiments, with system parameters as given in Table 1. Moreover the credibility of the proposed ME (infinite buffer case) and MRE (finite buffer case) solutions towards the analysis of Internet links carrying TCP traffic, outlined in the previous sections, is demonstrated against simulation via the same set of typical numerical experiments.

Table 1. Systems Parameters for the Infinite and Finite Buffer Cases

	Infinite Buffer System		Finite Buffer System				
ρ	<*n*>	*q*(0)	N	<*n*>	*p*(0)	*p*(*N*)	π
0.3	1.029	0.864	50	1.047	0.861	2.3E-4	9.1E-4
0.6	7.347	0.550	150	7.252	0.553	2.8E-5	6.1E-5
0.8	28.368	0.268	150	25.454	0.268	1.4E-3	2.3E-3
0.9	88.998	0.098	150	52.176	0.118	9.4E-3	1.6E-2

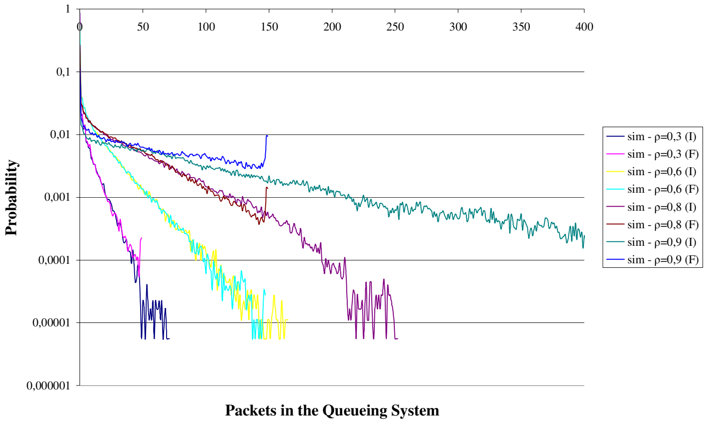


Fig. 2. *qld*s for the finite (F) and infinite (I) buffer cases obtained from simulation for ρ=0.3, 0.6, 0.8, 0.9

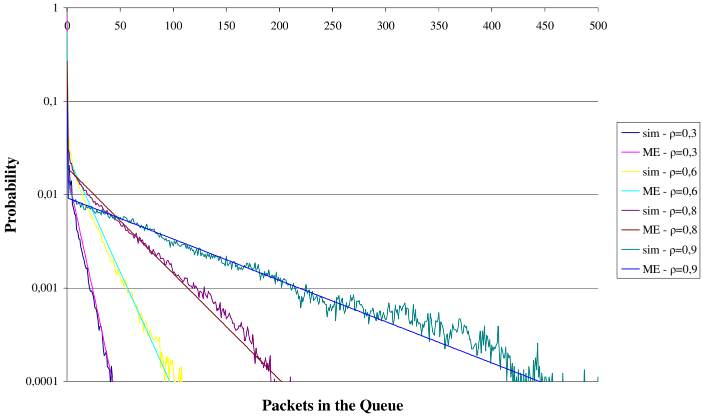


Fig. 3. *qld*s for the infinite buffer case obtained from simulation and the ME for ρ=0.3, 0.6, 0.8, 0.9

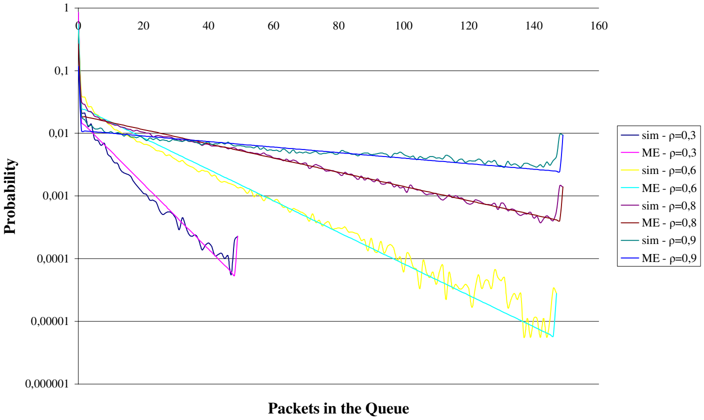


Fig. 4. *qld*s for the finite buffer case obtained from simulation and the ME for ρ=0.3, 0.6, 0.8, 0.9

Fig. 3 and Fig. 4 depict the qld for the infinite and finite buffer cases respectively for certain values of utilization, ρ. It is apparent, that the suggested approach manage to capture the behaviour of the system under various settings. In the finite buffer case, in particular, the MRE solution produces in accordance to the simulation a certain final peak at the full buffer probability, $p(N)$. An insight on the existence of this final peak is offered by this notion of 'parallel' distributions.

6 Conclusions

In this paper the information theoretic principles of Maximum Entropy and Minimum Relative Entropy were applied, as a method of inference towards the analysis of infinite and finite capacity systems. In particular, the principle of MRE is applied, as a method of inference, to the general problem of estimating finite buffer stationary qld, given, as a prior estimate, the qld of the corresponding infinite capacity buffer queue. Under the 'proportionality' relationship MRE is shown to provide with exact results for certain typical cases of queuing systems. It is also observed that certain real systems such as Internet links carrying TCP traffic exhibit this 'proportionality' relationship between finite and infinite systems. In the latter case, ME and MRE solutions manage to follow closely the behavior of the real systems, providing this way simple, cost effective, closed form expressions for the queue length distribution of the systems under study. Exhaustive experimentation showed the very good fit of the results with the outcome of simulation runs. Despite its simplicity, the methodology captures the behavior of the system under study both in the cases of finite and infinite buffers, and thus can easily be utilized for network management and design, capacity planning, and congestion control.

Further work is required to identify more subtle constraints that would give exact finite buffer qld for other types of queues with more complex interarrival and service time distributions, to analyze conditions for prescribed degrees of accuracy of approximate MRE solutions applicable to other cases of finite capacity queues where the 'proportionality' condition does not hold, and to extend the study to a network of internet links.

References

[1] Crovella, M.E., Bestavros, A., "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," In IEEE/ACM Transactions on Networking, 5(6):835–846, (1997).

[2] Grossglauser, M., Bolot, J., "On the Relevance of Long Range Dependence in Network Traffic," In IEEE/ACM Transactions on Networking, 7(5):629–640 (1999).

[3] Feldman, A., Gilbert, A., Huang, P., Willinger, W., "Data networks as cascades: Explaining the multifractal nature of Internet Wan tra.c," In ACM SIGCOMM '98, pp.42–55, Vancouver, Canada (1998).

[4] Willinger, W., Paxson, V., "Where Mathematics meets the Internet," Notices of the American Mathematical Society, 45(8):961–970 (1998).

[5] Erramilli, A., Narayan, O., Willinger, W., "Experimental queuing analysis with long-range dependent packet traffic," In IEEE/ACM Transactions on Networking, 4(2):209–223 (1996).

[6] Erramilli, A., Narayan, O., Neidhardt, A., "Performance Impacts of Multi-Scaling in Wide Area TCP/IP Traffic" In IEEE INFOCOM '00, Tel Aviv, Israel (2000).

[7] Ribeiro, V., Riedi, R., Crouse, M., Baraniuk, R., "Multiscale Queuing Analysis of Long-Range-Dependent Network Traffic," In IEEE INFOCOM '00, Tel Aviv, Israel (2000).

[8] Vanichpun, S., Makowski, A., "Positive correlations and buffer occupancy: Lower bound via supermodular ordering," In IEEE INFOCOM '02, New York, NY (2002).

[9] Cardwell, N., Savage, S., Anderson, T., "Modeling TCP Latency," In IEEE INFOCOM '00, Tel Aviv, Israel (2000).

[10] Fredj, S.B., Bonald, T., Proutiere, A., Regnie, G., Roberts, J., "Statistical Bandwidth Sharing: A Study of Congestion at Flow Level," In ACM SIGCOMM '01, pp.111-122, San Diego, USA (2001).

[11] Barakat, C., Thiran, P., Iannaccone, G., Diot C., Owezarski, P., "A flow-based model for Internet backbone traffic," In ACM Internet Measurement Workshop, Marseille, France (2002).

[12] Garetto, M., Towsley, D., "Modeling, Simulation and Measurements of Queuing Delay under Long-tail Internet Traffic", In SIGMETRICS'03, pp. 47-57, San Diego, USA (2003).

[13] Appenzeller, G., Keslassy, I., McKeown, N., "Sizing router buffers", In ACM SIGCOMM '04, pp. 281-292, USA, August/September 2004.

[14] Benes, V.E., "Mathematical Theory of Connecting Networks and Telephone Traffic", Academic Press, New York (1965).

[15] Ferdinand, A.E., "A Statistical Mechanical Approach to Systems Analysis", IBM Journal of Research and Development, vol. 14, pp.539-547 (1970).

[16] Pinsky, E., Yemini, Y., "A Statistical Mechanics of Some Interconnection Networks", Performance '84, North-Holland, pp. 147-158 (1984).

[17] Jaynes, E.T., "Information Theory and Statistical Mechanics I", Physical Review, vol.106, pp.620-630 (1957).

[18] Jaynes, E.T., "Information Theory and Statistical Mechanics II", Physical Review, vol.108, pp.171-190 (1957).

[19] Shore, J.E., Johnson, R.W., "Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum-Cross Entropy", IEEE Trans. on Information Theory, vol.IT-26, pp.26-37 (1980).

[20] Shore, J.E., Johnson, R.W., "Properties of Cross Entropy Minimisation", IEEE Trans. on Information Theory, vol.IT-27, pp.472-482 (1981).

[21] Kouvatsos, D.D., "Maximum Entropy and the G/G/1/N Queue", Acta Informatica, vol.23, pp.545-565 (1986).

[22] Kouvatsos, D.D., "A Maximum Entropy Analysis of the G/G/1 Queue at Equilibrium", Journal of Oper. Research Society, vol.39, pp.183-200 (1988).

[23] Skianis, C., Kouvatsos, D.D., "Arbitrary Open Queueing Networks with Server Vacation Periods and Blocking", Special Issue on Queueing Networks and Blocking, Annals of Operations Research 79, pp. 143-180 (1998).

[24] McCanne, S., Floyd, S., "Ns-2 network simulator", http://www.isi.edu/nsnam/ns/

Characterization of the Burst Aggregation Process in Optical Burst Switching

Xenia Mountrouidou and Harry G. Perros

Computer Science Department,
North Carolina State University,
Raleigh, NC 27695, USA
{pmountr, hp}@csc.ncsu.edu

Abstract. We describe an analytic approach for the calculation of the departure process from a burst ggregation algorithm that uses both a timer and maximum/minimum burst size. The arrival process of packets is assumed to be Poisson or bursty modelled by an Interrupted Poisson Process (IPP). The analytic results are approximate and validation against simulation data showed that they have good accuracy.

1 Introduction

An important design aspect of an OBS network is the burst aggregation process performed at the edge nodes. This process concentrates upper layer packets which are then transmitted optically over the OBS network. In view of this, the burst aggregation strategy defines the burst arrival process to the OBS network. This process depends on the parameters of the aggregation process, and so far it has not been adequately studied. However, it is important that the burst arrival process to an OBS network is well characterized if we are to understand better the performance of OBS networks.

The main parameters of a burst assembly algorithm are a timer and the maximum and minimum burst size. When the timer expires, the edge node assembles a burst that consists of packets in the edge's packet queue that have the same destination. This procedure takes place in the electrical domain, and the resulting bursts are transmitted in the optical domain. If the arrival rate at an edge node is very high, then each time the timer expires, there may be a large number of packets waiting in the packet queue. This would lead to large data bursts if all packets are assembled into a single burst. Large bursts decrease the performance in the OBS network, since they occupy the resources for long intervals. As a result, they block other bursts thus leading to a high burst loss probability [1]. In order to avoid this problem, a maximum burst size is used to bound the size of a burst. Another drawback of the burst assembly algorithm, if it is driven entirely by a timer, is that bursts can be very small if the arrival rate to the edge node is very low. This results to high overheads, since the OBS network has to set up a path for each burst. The solution to this problem is to use a lower bound for the burst size. If this lower limit is not reached, when the timer expires, then the burst is not transmitted.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 752–764, 2006.

Various algorithms have been proposed to aggregate packets into bursts. Most of these assembly algorithms use either an assembly timer or a maximum and minimum burst length or both as a way of creating bursts. Let T be the length of the timer, B_{max} the maximum burst length and B_{min} the minimum burst length. We can classify the assembly algorithms into the following three categories:

- **Time-based aggregation algorithms:** In this case a fixed-threshold T is used to create a burst. In some implementations, a minimum length B_{min} is required [2]. If the burst is shorter than B_{min} then padding is used to increase the length to B_{min}. The shortcoming of the time-based assembly algorithm is that, under heavy traffic load, the number of packets that are gathered until the timer expires may be high, thus resulting to large bursts.
- **Burst-length based aggregation algorithms:** In this case, the burst is sent out as soon as the burst length exceeds a given maximum burst length B_{max}. Thus, the packets are buffered until the total size reaches the maximum threshold. The main disadvantage of this algorithm is that it does not constraint the waiting time of the packets in the packet queue. Therefore, when the traffic is low, waiting time may be large.
- **Time and burst-length based burst aggregation algorithms:** The disadvantages of the aggregation algorithms based on a timer or a maximum burst length can be overcome using a combination of a timer and maximum and minimum burst lengths. In this case, the packets are buffered until the timer expires. Then, we compare the total size of the packets in the queue with the upper and lower limits, B_{max} and B_{min}. If the size is less than B_{min}, then we keep the packets in the packet queue until the next aggregation period, i.e. until the next time when the timer expires. If the size is greater than B_{min} but less than B_{max} we aggregate all the packets in one burst. If the size is greater than B_{max}, we make one burst of maximum size and then we repeat this process with the remaining bits.

We note that an adaptation scheme has to be created which will assemble packets into bursts at the transmitter's side, and correctly recover these packets at the receiver's side. There are several examples of adaptation schemes, such as the schemes used for the formation of AAL 5 PDU and AAL 2 CPS packets in ATM networks. In OBS depending upon the adaptation scheme, a packet may straddle over two successive bursts. Alternatively, a burst may be allowed to exceed a maximum burst size, so that the last packet is included in its entirety. In this paper we assume that the maximum burst size is strictly enforced, and therefore a packet may straddle over two successive bursts.

The burst aggregation process has been studied in [3], [4], [5], [6] and [2]. Papers [4], [5], [6] and [2] study the effect of burst aggregation algorithms on the self-similarity characteristics of the input traffic. [3] gives an analytical method to calculate the aggregated burst size for various algorithms, and assuming Poisson arrivals of packets to the edge node. The authors did not consider the aggregation algorithm that uses both a timer and a maximum/minimum burst size analyzed in this paper. In this paper we obtain analytically the distribution of the number of bursts created by the aggregation algorithm which uses both a timer and a maximum/minimum burst size. We first assume that the arrival of packets to the edge node is Poisson, and then we extend the analysis to the case where packets arrive according to an Interrupted Poisson Process (IPP). This process models bursty traffic such as video, voice or data.

The rest of this paper is organized as follows. In Section 2 we obtain analytically the distribution of the number of bursts created at each aggregation period assuming Poisson arrivals. In Section 3 we extend these results to IPP arrivals. The results obtained in this paper are approximate and in Section 4 we compare our analytical results with simulation data. Finally, Section 5 gives the conclusions.

2 The Case of Poisson Arrivals

We consider an edge node which receives packets in the electronic domain and transmits them to destination edge nodes optically over an OBS network. The arriving packets are queued to different packet queues, each associated with a different destination edge node. We only consider a single packet queue in which packets are queued for a specific destination. We assume that the arrival process of packets to the queue is Poisson with a rate of λ. Packet size is exponentially distributed with a mean of $1/b$ *bytes*. We recall that the length of the aggregation period, i.e. the time after which the timer expires, is T.

Since packets arrive in a Poisson fashion, the probability that n packets arrive within T is: $P[X = n] = e^{-\lambda T} \frac{(\lambda T)^n}{n!}$. Therefore, the pdf of the number of bytes B that arrive during the i^{th} aggregation period $((i-1)T, iT]$ is:

$$f_B(x) = \sum_{n=1}^{\infty} P[X = n] f_{S_n}(x), \tag{1}$$

where $f_{S_n}(x)$ is the probability that the total number of bytes associated with n packets is x. This is obtained by convoluting n i.i.d. exponentially distributed variables, which in fact is the pdf of an n-stage Erlang distribution [7], given by: $f_{S_n}(x) = \frac{b(bx)^{n-1}e^{-bx}}{(n-1)!}$. Thus, the pdf of the number of bytes that arrive during the i^{th} aggregation period is:

$$f_B(x) = \sum_{n=1}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n b(bx)^{n-1} e^{-bx}}{n!(n-1)!} \tag{2}$$

The cumulative distribution function (cdf) of the number of bytes in the packet queue at the end of the period T is:

$$F_B(x) = \int_0^{\infty} f_B(x)$$

where $f_B(x)$ is given by (2). Using $F_B(x)$ we can calculate the probability of the number of bursts that are formed at the end of each period T. For instance, the number of bytes x has to be within the interval $[0,\ B_{min}-1]$ in order to have zero bursts, it has to be within $[B_{min},\ B_{min}+B_{max}-1]$ in order to have one burst, and in general it has to be within the interval $[B_{min}+(k-1)B_{max},\ B_{min}+kB_{max}-1]$ in order to have k bursts. That is:

$$P[k = 0\ bursts] = \int_{0}^{B_{min}-1} f_B(x) \qquad (3)$$

$$P[k\ bursts] = \int_{B_{min}+(k-1)B_{max}}^{B_{min}+kB_{max}-1} f_B(x),\ k >= 1 \qquad (4)$$

Expression 2 is quite difficult to work with, and whenever possible it is approximated by a simple mixture of exponential distribution as described below. As will be seen in order to do this, we need the first three moments of the number of bytes in the packet queue at the end of each aggregation period. We note that the number of bytes that arrive during period T is a random sum of exponentially distributed variables. Therefore, the moment generating function (MGF) of the number of bytes is [8]:

$$M_B(t) = M_N(ln(M_S(t))) \qquad (5)$$

where $M_B(t)$ is the MGF of the number of bytes during interval $((i-1)T, iT]$, $M_N(t)$ is the MGF of the number of packets N, and $M_S(t)$ the MGF of the packet size S. Thus, we have: $M_N(t) = e^{\lambda T(e^{ln(M_S(t))}-1)}$, $M_S(t) = \frac{b}{b-t}$ and therefore:

$$M_B(t) = e^{\lambda T(\frac{b}{b-t}-1)} \qquad (6)$$

At the end of each aggregation period there may be a residual number of bytes r which are not transmitted in a burst because of the condition that a burst has to be at least greater than B_{min}. We have found empirically that if $B_{min} <<< B_{max}$, then the residual number of bytes is typically zero. On the other hand, if B_{min} is close to B_{max}, then we have observed that the residual number of bytes is uniformly distributed within $[0,\ B_{min})$. For instance, if $B_{min} = 16\ Kbytes$ and $B_{max} = 112\ Kbytes$, then there is a high probability that the last burst will be larger than 16 Kbytes, which means that the residual number of bytes will be zero. However, if we set $B_{min} = 85Kbytes$ then there is a high probability that the last burst will not be greater than B_{min}, which means that it will not be transmitted out. This remainder can be safely assumed that it is uniformly distributed within $[0,\ B_{min})$.

In view of the above empirical observations, we distinguished two cases. If $B_{min} <<< B_{max}$, then we assume that there is zero left over bytes from the previous aggregation period, in which case the pdf of $f_B(x)$ and its MGF $M_B(t)$ are given by expressions (4) and (8). If B_{min} is close to B_{max}, then the pdf $f_X(x)$ of the number of bytes at the end of an aggregation period is given by the convolution of $f_B(x)$ and $f_r(x)$. We have: $f_X(x) = f_B(x) * f_r(x)$. Therefore, the MGF of $f_X(x)$, $M_X(t)$ is given by ([9]):

$$M_X(t) = M_B(t)M_r(t)\ or\ M_X(t) = \begin{cases} e^{\lambda T(\frac{b}{b-t}-1)}\frac{e^{tB_{min}-1}}{tB_{min}} & \text{if } t > 0 \\ e^{\lambda T(\frac{b}{b-t}-1)} & \text{if t} = 0 \end{cases} \qquad (7)$$

We can now calculate the moments of the distribution of the number of bytes available in an aggregation period for both models. In the first model, where we do not include the residual number of bytes , we have:

$$m_1 = M'_B(0) = \frac{\lambda T}{b}, \tag{8}$$

$$m_2 = M''_B(0) = \frac{\lambda T}{b^2}(2 + \lambda T) \tag{9}$$

$$m_3 = M_B^{(3)}(0) = \frac{\lambda T}{b^3}[(2 + \lambda T)(3 + \lambda T) + \lambda T] \tag{10}$$

In the second model, where we include the residual of the number of bytes, we have:

$$m_1 = M'_X(0) = \frac{\lambda T}{b} + \frac{B_{min}}{2}, \tag{11}$$

$$m_2 = M''_X(0) = \frac{\lambda T}{b^2}(2 + \lambda T) + \frac{B_{min}^2}{3} + \frac{\lambda T B_{min}}{b} \tag{12}$$

$$m_3 = M_X^{(3)}(0) = \frac{\lambda T}{b^3}[(2+\lambda T)(3+\lambda T)+\lambda T] + \frac{B_{min}^3}{4} + \frac{3\lambda T B_{min}}{b}\left[\frac{1}{2b}(2+\lambda T) + \frac{B_{min}}{3}\right] \tag{13}$$

From the first three moments of these different models, we see that the number of bytes that arrive within a period T and the residual from the previous period are independent. This is because of the way we calculated the total number of bytes available at the end of period T.

Using the three moments, we can now approximate the pdf $f_B(x)$ or $f_X(x)$ of the total number of bursts in the packet queue at the end of a period T by a two-stage Coxian, C_2 [10]. For this we set the first three moments, m_1, m_2, m_3 equal to the first three moments of C_2 with parameters (μ_1, μ_2, α). The three moment fit, can be used if $3m_2^2 > 2m_1m_3$ and $c^2 > 1$, where c^2 is the squared coefficient of variation. Alternatively a two moment fit can be used if the condition $3m_2^2 > 2m_1m_3$ does not hold or $0.5 < c^2 < 1$. The pdf of a C_2 is given by the expression:

$$f_Y(y) = (1 - \alpha)\mu_1 e^{-\mu_1 y} + \alpha\left(\frac{\mu_1\mu_2}{\mu_2 - \mu_1} e^{-\mu_1 y} + \frac{\mu_1\mu_2}{\mu_1 - \mu_2} e^{-\mu_2 y}\right) \tag{14}$$

where in the case of the three-moment fit: $\mu_1 = \frac{L+(L^2-4K)^{1/2}}{2}$, $\mu_2 = L - \mu_1$ *and* $\alpha = ((\mu_1 m_1) - 1)$, $K = \frac{6m_1 - 3(m_2/m_1)}{((6m_2^2)/(4m_1) - m_3}$, $L = 1/m_1 + \frac{m_2 K}{2m_1}$ and in the case of the two-moment fit: $\mu_1 = \frac{2}{\mu_1}$, $\mu_2 = \frac{1}{\mu_1 c^2}$ *and* $\alpha = \frac{1}{2c^2}$. Using the C_2 pdf we can easily calculate the cumulative distribution $F_X(x)$ and from there the probability of creating k bursts at the end of each period T (see equations 3, 4).

When $c^2 < 0.5$, we can fit an Erlang distribution or a generalized Erlang distribution (see [10]). However we observed empirically, that the number of bytes in the packet queue at the end of an aggregation period has a small variability. In view of this, we have found that it is sufficient to evaluate $f_B(x)$, given by equation 2, for only a small range of values of n. That is, we limit the sum to:

$$f_B(x) = \sum_{n=avgNumPacks-10\sigma}^{avgNumPacks+10\sigma} \frac{e^{-\lambda T}(\lambda T)^n b(bx)^{n-1}e^{-bx}}{n!(n-1)!} \tag{15}$$

where $avgNumPacks = \lambda T$, is the average number of packets that arrive during a period T, and $\sigma = \sqrt{\lambda T}$ is the variance of the number of packets that arrive during T. From 15 we can then numerically compute the cumulative distribution of the pdf $f_B(x)$ or $f_X(x)$ and subsequently the probability of having k bursts, where $k \geq 0$ (see equations 3, 4).

Due to the limited variability of the number of bytes in T, we have also found that the following approximation gives good results:

$$P[k\ bursts] = \frac{m_1/B_{max}}{\lceil m_1/B_{max} \rceil},\ P[(k-1)\ bursts] = 1 - P[k\ bursts] \tag{16}$$

where $k = m_1/B_{max}$.

3 The Case of IPP Arrivals

The IPP is a modified Poisson process. It is similar to a Markov Modulated Poisson Process with two states (MMPP2). The main difference between IPP and MMPP2 is that the arrival rate in the second state of the MMPP2 is zero, which means there are no arrivals in this state [11]. An IPP is an ON/OFF process, where the ON and OFF periods are exponentially distributed with rates σ_1 and σ_2 respectively. We also define the vector π: $\boldsymbol{\pi} = (\pi_1, \pi_2) = \frac{1}{\sigma_1+\sigma_2}(\sigma_2, \sigma_1)$ which gives the average duration of the ON and OFF periods. During the ON period there are Poisson arrivals with rate λ, and during the OFF period there are no arrivals. This is a very useful model for data/voice and video transfers over the Internet, where bursty arrivals of packets occur for a period of time followed by an idle interval. We assume that the packet sizes are exponentially distributed with an average size $1/b$ bytes. The IPP process is also characterized by the squared coefficient of variation, c^2_{IPP}, of the packet interarrival time, that measures the burstiness of the arrival process, given by the expression: $c^2_{IPP} = 1 + \frac{2\lambda\sigma_1}{(\sigma_1+\sigma_2)^2}$. From this equation we can observe that the value of c^2_{IPP} is affected by the duration of the ON and OFF periods. We also define the quantity: $average\ transmission\ rate = transmissionSpeed\frac{\sigma_2}{\sigma_1+\sigma_2}$ The *average transmission rate* depends on the transmission speed, it is in fact the transmission speed within the ON period. In our implementation, we set $1/\lambda = \frac{average\ packet\ size}{transmissionSpeed} = \frac{1/b}{10Gbps}$, where $1/b = 500\ bytes$. Thus, $1/\lambda$ is the time needed to transmit one packet during the ON period. Given the c^2_{IPP} and the *average transmission rate* we calculate the ON and OFF periods : $\frac{1}{\sigma_1}$ and $\frac{1}{\sigma_2}$ respectively.

In this section we calculate the pdf of the number of bursts during an aggregation period assuming that packets arrive in IPP fashion. We follow the same approach as in the case of Poisson arrivals. We first calculate the MGF of the number of bytes that arrive during a period T. Similarly to the Poisson arrival case, we have a random sum of packets whose size is exponentially distributed, and therefore we use equation 5: $M_B(t) = M_N(ln(M_S(t)))$ where $M_B(t)$ is the MGF of the number of bytes during interval $((i-1)T, iT]$, $M_N(t)$ is the MGF of the number of packets N, and $M_S(t)$ the MGF of the packet size. The MGF of the number of packets that arrive during a period T is obtained as follows.

Let: $P_{ij} = Prob\{N_t = n, J_t = j | N_0 = 0, J_0 = i\}$ be the probability that N_t arrivals occur during $(0, t]$ given that at time 0 there were 0 arrivals and the IPP was in state $J_0 = i$ and at time t the IPP was in state $J_t = j$. The z-transform of P_{ij} [11] is: $P^*(z, t) = e^{(Q-(1-z)\Lambda)t}$ where Q is the infenitesimal generator of the IPP and Λ the matrix of arrival rates, i.e.

$$Q = \begin{pmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{pmatrix}, \Lambda = \begin{pmatrix} \lambda & 0 \\ 0 & 0 \end{pmatrix}$$

Now we can use this z-transform to form the generating function of the number of packets. We know that the MGF of a discrete random variable x, is its z-transform when $z = e^s$. Therefore, $M_X(s, t) = P^*(e^s, t)$, where $P^*(z, t)$ the z-transform of function $P(n, t)$. Thus, the MGF of the number of packets that arrive during $(0, t]$ is: $M_N(s, t) = e^{(Q-(1-e^s)\Lambda)t}$. Substituting s with $ln(M_S(-s))$, where $M_S(s) = \frac{b}{b+s}$, is the Laplace transform of the exponentially distributed packet size, we obtain:

$$M_B(s, t) = e^{(Q-(1-\frac{b}{b-s})\Lambda)t} \tag{17}$$

We can now calculate the first two moments $m_{1,IPP}$ and $m_{2,IPP}$ of the number of bytes that arrive during a period T, by setting $t = T$. We have:

$$m_{1,IPP} = \mathbf{e}^T \mu(s) = \mathbf{e}^T M'(T)\mathbf{e} \tag{18}$$

where: $M'(T) = \left[\frac{\partial}{\partial s} M_B(s, T)\right]_{s=0}$ and $\mathbf{e}$ is a column vector of 1's and $\mathbf{e}^T$ a row vector of 1's. In order to differentiate the MGF we use the eigenvalue decomposition of the matrix exponential given in 17. The eigenvalue decomposition always exists for this MGF. We have: $e^{At} = Pe^{Dt}P^{-1}$ where $A = (Q - (1 - \frac{b}{b-s})\Lambda)t$ D is the diagonal matrix of the eigenvalues of A, P is the matrix composed of eigenvectors and P^{-1} the inverse matrix of P. After differentiating and using the chain rule we get:

$$M'(T) = \frac{\partial e^{AT}}{\partial s} = \frac{\partial P}{\partial s} e^{DT} P^{-1} + Pe^{DT} \frac{\partial P^{-1}}{\partial s} + TPe^{DT} \frac{\partial D}{\partial s} P^{-1} \tag{19}$$

Substituting the above in equation 18 we have:

$$m_{1,IPP} = \pi_1 \frac{\lambda T}{b} \tag{20}$$

The above expression is intuitively obvious, as it is the mean duration of the ON period π_1 multiplied by the mean number of bytes that arrive during this period $\frac{\lambda T}{b}$. The second moment is given by:

$$m_{2,IPP} = \mathbf{e}^T \mu_2(s) = \mathbf{e}^T M^{(2)}(T)\mathbf{e} \tag{21}$$

where the second derivative $M^{(2)}(T) = \left[\frac{\partial^2}{\partial s} M_B(s,T)\right]_{s=0}$ is obtained by applying the chain rule to equation 19. We have:

$$\begin{gathered} M^{(2)}(T) = \frac{\partial^2 P}{\partial s} e^{DT} P^{-1} + 2T \frac{\partial P}{\partial s} e^{DT} \frac{\partial D}{\partial s} P^{-1} + 2 \frac{\partial P}{\partial s} e^{DT} \frac{\partial P^{-1}}{\partial s} \\ +2TPe^{DT} \frac{\partial D}{\partial s} \frac{\partial P^{-1}}{\partial s} + Pe^{DT} \frac{\partial^2 P^{-1}}{\partial s} + T^2 P e^{DT} \left(\frac{\partial D}{\partial s}\right)^2 P^{-1} \\ +TPe^{DT} \frac{\partial^2 D}{\partial s} P^{-1} \end{gathered}$$

Now we can use the two moments to approximate the pdf $f_B(x)$ of the number of bytes that arrive during a period T, with a C_2 distribution as in the previous section.

If $c^2 < 0.5$ then we used the generalized Erlang k approximation, where: $\frac{1}{k} \leq c^2 \leq \frac{1}{k-1}$ In the generalized Erlang k with probability a, after the first exponential phase the service continues for the rest of the $k-1$ stages or it ends with probability $1-a$. Probability a is given by [10]:

$$1 - a = \frac{2kc^2 + k - 2 - \sqrt{k^2 + 4 - 4kc^2}}{2(c^2+1)(k-1)} \tag{22}$$

and service rate: $\mu = \frac{1+(k-1)a}{m_1}$

The probability of having k bursts at the end of an aggregation period T is:

$$P[k\ bursts] = \int_{(k-1)B_{max}}^{kB_{max}-1} f_Y(y),\ k >= 1$$

where $f_Y(y)$ is the pdf of the C_2 or the generalized Erlang pdf as above. Notice that $B_{min} = 0$ in this model, since there are no residual bytes included. The case of $B_{min} > 0$ can be modelled as in the previous section.

Finally, we note that the number of bytes that arrive during each interval T is approximated by the number of bytes that arrive in $(0, t]$. In reality, the IPP state at the beginning of each interval T is the state at the end of the previous period T. In our model, we approximate this by assuming that the IPP is at state i at the beginning of the interval T and then uncondition on this event.

4 Numerical Results

In this section we compare our approximate analytic results to simulation data for both Poisson and IPP arrivals. A simulation program was writen in C to simulate the behavior of the burst aggregation algorithm under study, with Poisson

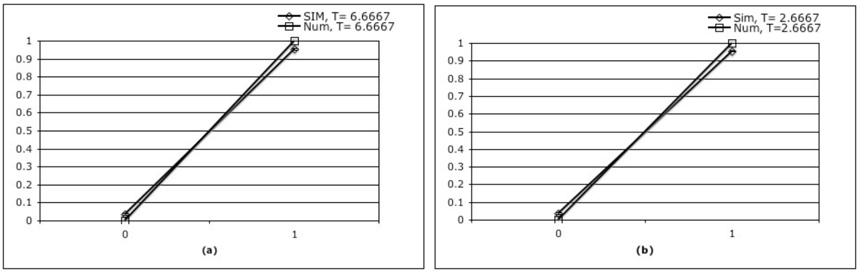


Fig. 1. Probability distribution of the number of bursts. Poisson Arrivals, no residual, $B_{min} = 0$.

and IPP arrivals. 95% confidence intervals were computed using the method of batch means. The number of batches was fixed to 30 and each batch consisted of 100,000 aggregation periods T. The confidence intervals were extremely small and they are not discernible in the figures.

4.1 Poisson Arrivals

Figures 1, 2 and 3 give approximation and simulation results of the probability distribution of the number of bursts for the case where $c^2 > 0.5$. In this case the probability distribution is computed using the fitted C_2 distribution. Figures 1 (a) and 1 (b) give results for $T = 6.6667$ and $T = 2.6667$ respectively. $B_{min} = 0$, which means that the probability of having zero bursts is 0, $B_{max} = 112\ Kbytes$. The arrival rate λ was 2.5 $packets/\mu sec$ and 0.5 $packets/\mu sec$ for 1 (a) and 1 (b) respectively. The average packet size $1/b$ is obtained from the expression: $1/\lambda = \frac{8(1/b)\ (bits)}{transmission\ speed\ (Gbps)}$, where the transmission speed is 10 Gbps. We observe that the results are quite accurate. In the case where $B_{min} = 16\ Kbytes$, we have observed that our approximation is slightly affected by the residual bytes. This is because in the case where $B_{min} = 16\ Kbytes$ we may have residual bytes but this model does not include them since they are very few and usually 0. This is why we have a variation from the simulation results.

Figures 2 (a) and 2 (b) give the analytical and simulation results of the probability distribution of the number of bursts when B_{min} is close to B_{max}. In this case, we include the residual bytes from the previous aggregation period in our calculation. $B_{max} = 200\ Kbytes, B_{min} = 150\ Kbytes$, the average packet size $1/b = 125\ Kbytes$ and $transmissionSpeed = 1\ Tbps$. These parameters could be meaningful in very high speed networks with dedicated connections where large file transfers may occur, such as in a Grid environment. In Figure 2 (a) and in Figure 2 (b) $T = 2.05549\ \mu sec$ and $T = 2.91172\ \mu sec$. Our approximation is very accurate in this case, only a slight variation is observed for a low number of bursts that could be justified since the assumption that the residual bytes are uniformly distributed in $[0,\ B_{min})$ is not always accurate. This assumption is more accurate when B_{min} is higher (180 Kbytes) and therefore the difference $B_{max} - B_{min}$ is smaller, as can be viewed in Figures 3 (a) and 3 (b).

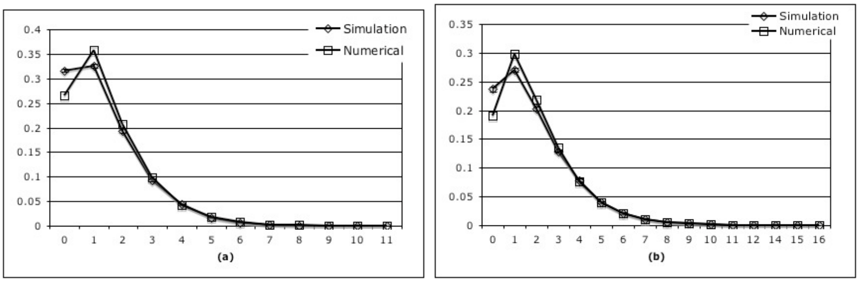


Fig. 2. Probability distribution of the number of bursts. Poisson Arrivals, with residual, $B_{min} = 150\ Kbytes$.

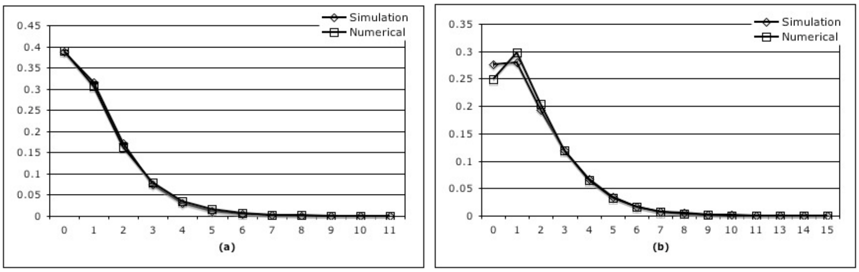


Fig. 3. Probability distribution of the number of bursts. Poisson Arrivals, with residual, $B_{min} = 180\ Kbytes$.

Figure 4 gives results for a case where $c^2 < 0.5$. As mentioned above, we analyze this case, either by computing equation 2 for a limited range of values, or using the approximation given in (18). In this case the approximate results match the simulation data. The latter approach is very fast but it does not give accurate results in all cases. The former method is much slower, but gives very accurate results.

Figures 4 (a), 4 (b), 4 (c) and 4 (d) give the analytic results obtained using both methods and the simulation results for $T = 128,\ 256,\ 512$ and $1024\ \mu sec$ respectively. No residual is included and $B_{min} = 16\ Kbytes$. The transmission speed was 10 Gbps, the average packet size, based on IP packets, was 500 bytes. In the case where $T = 128,\ 256,\ 512\ \mu sec$ both analytic methods give accurate results. When $T = 1024\ \mu sec$ the aggregation period is high and the variability in the number of bursts increases. Thus in this case the approximation method is not accurate. If the residual bytes are included then we have observed that both our analytic models give almost the same results as the simulation model for a variable aggregation period T. We note that when $c^2 < 0.5$ the distribution of the available number of bytes at the end of each aggregation period is almost constant. This explains why the probability distribution of the number of bursts is almost constant. The result can prove useful in traffic engineering as it may simplify the architecture.

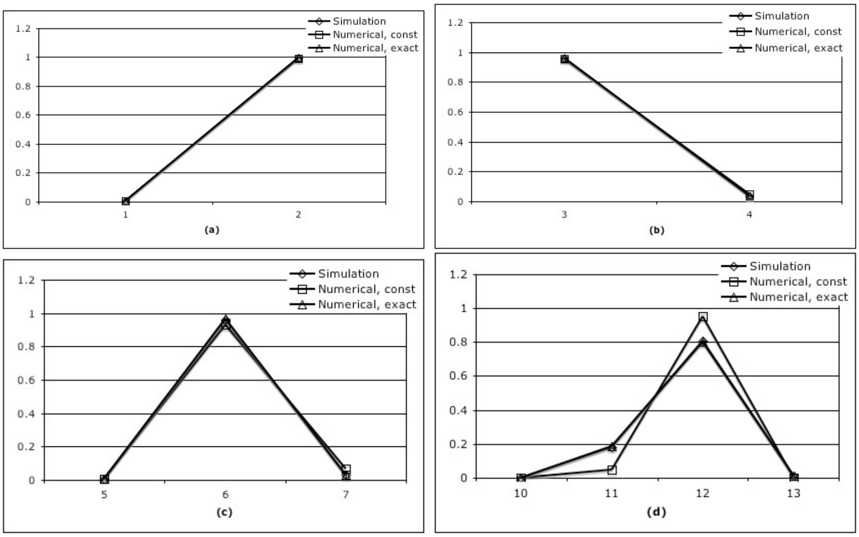


Fig. 4. Probability distribution of the number of bursts. Poisson Arrivals, no residual, $B_{min} = 16\ Kbytes$.

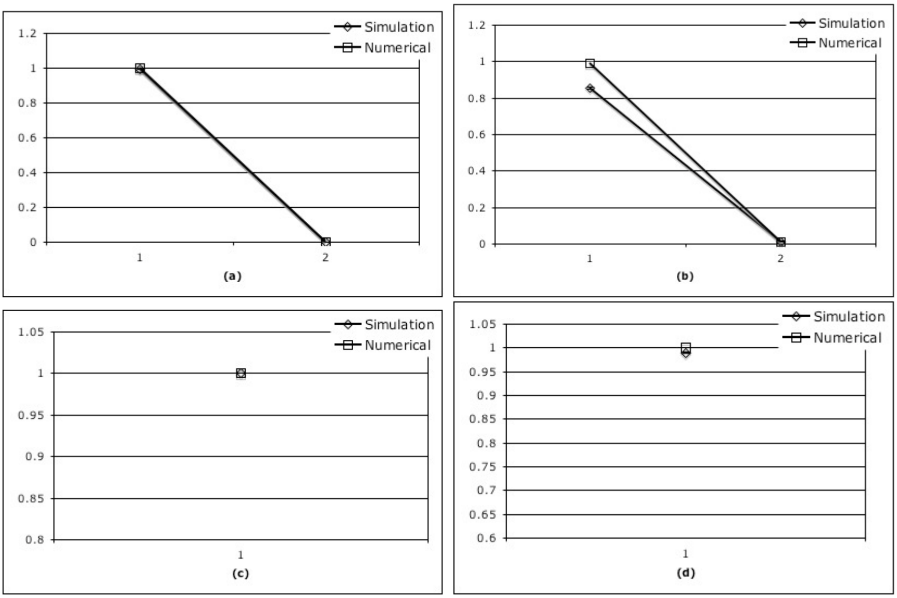


Fig. 5. Probability distribution of the number of bursts. IPP arrivals, no residual, $B_{min} = 0\ bytes, Average\ Arrival\ Rate = 1\ Gbps$.

4.2 IPP Arrivals

The results that are given in Figure 5 were obtained under the following assumptions: *transmission speed* = 10 Gbps, *average transmission rate* = 1 Gbps, $c^2_{IPP} = 5$, $B_{min} = 0$, $B_{max} = 16$ or 112 Kbytes. average packet size $1/b = 500$ bytes, and arrival rate during ON period: $\lambda = 2.5$ *packets/µsec*. In this case, $c^2 > 0.5$ and we use the C_2 fit.

Figures 5 (a) and 5 (b) show the probability of having k bursts $B_{max} = 16$ *Kbytes* and the aggregation period is $T = 16,\ 32$ *µsec* respectively. In Figures 5 (c) and 5 (d) we increase the B_{max} to 112 Kbytes. There is almost no difference between the simulation results and the numerical results.

5 Conclusions

The burst aggregation process defines to a large extent the burst arrival process to the OBS network. This burst arrival process has not as yet been adequately studied. However, it is important that it is well characterized if we are to understand better the performance of the OBS network. In this paper, we have obtained analytically the probability distribution of the number of bursts created by an aggregation algorithm that uses a timer and a minimum and maximum burst size. The analytical results are approximate but they seem to have good accuracy.

Acknowledgements. We would like to thank Boldea Otilia for her helpful comments in the IPP analysis.

References

1. Xu, L., Perros, H.: Performance analysis of an ingress optical burst switching node. (Submitted. http://www.csc.ncsu.edu/faculty/perros/Xu5.pdf)
2. Yu, X., Li, J., Cao, X., Chen, Y., Qiao, C.: Traffic statistics and performance evaluation in optical burst switched networks. Journal of lightwave technology **22**(12) (2004) 2722–2738
3. de Vega Rodrigo, M., Gotz, J.: An analytical study of optical burst switching aggregation strategies. In: Broadnets 2004, IEEE (2004)
4. Hu, G., K., D., Gauger, C.: Does burst assembly really reduce the self-similarity. In: In Proc. of the Optical Fiber Communication Conference (OFC). (2003)
5. Xiong, Y., Vandenhoute, M., Cankaya, H.: Control architecture in optical burst switched wdm networks, in ieee jsac. Volume 18. (2000) 1838–1851
6. Xue, F., Yoo, S.J.B.: Self-similar traffic shaping at the edge router in optical packet-switched networks. In: In Proc. of IEEE International Conference on Communications. (2002)
7. Viniotis, Y.: Probability and Random Processes for Electrical Engineers. McGraw-Hill (1997)
8. Yates, R.D., Goodman, D.J.: Probability and Stochastic Processes. John Wiley and Sons (1999)

9. Kleinrock, L.: Queueing Systems, Volume I: Theory. John Wiley and Sons (1975)
10. Perros, H.G.: Queueing Networks with Blocking, Exact and Approximate Solutions. Oxford University Press (1994)
11. Fischer, W., Meier-Hellstern, K.: The Markov-Modulated Poisson Process (MMPP) cookbook. Performance Evaluation **18** (1992) 149–171

Improving Bandwidth Efficiency in a Multi-service Slotted Dual Bus Optical Ring Network

Mohamad Chaitou, Gérard Hébuterne, and Hind Castel

Institut National des Télécommunications,
9 rue Charles Fourrier, 91011 Evry cedex, France
Tel.: +33 (0) 1 60 76 46 91; Fax: +33 (0) 1 60 76 42 91
{Mohamad.Chaitou, Gerard.Hebuterne, Hind.Castel}@int-evry.fr

Abstract. The paper proposes two IP packet aggregation techniques, called DAT (Deterministic Aggregation Technique) and WATT (Work-conserving Aggregation Technique with Timer), to adapt IP traffic to a multi-service optical slotted network. To perform aggregation, IP packets belonging to different classes of service (CoS) are polled according to the strict priority (SP) or to an hybrid version of the probabilistic priority (PP) scheduling discipline originally proposed in [1, 2]. An approximate analytical model is given in the case of DAT under the SP discipline. In addition, extensive simulations are used to study the impact of self-similar traffic on the aggregation processes. Finally, performance comparisons between the aggregation techniques and the standard approach (where no aggregation is performed) are carried out in the context of a slotted dual bus optical ring network (SDBORN) which is a candidate viable solution for metropolitan area networks (MAN).

keywords: Packet aggregation, slotted rings, bandwidth efficiency.

1 Introduction

In recent years, considerable research has been devoted to design IP full optical backbone networks, based on Wavelength Division Multiplexing (WDM) technology, in order to relieve the capacity bottleneck of classical electronic-switched networks. In a long-term scenario, optical packet switching (OPS), based on fixed-length packets and synchronous node operation, can provide a simple transport platform based on a direct IP over WDM structure which can offer high bandwidth efficiency, flexibility, and fine granularity. In order to support several CoS and to adapt the asynchronous and variable size behavior of IP traffic to OPS networks, the aggregation of IP packets at the interface of optical networks presents an efficient solution among few other proposals in literature (e.g., [3]). This is because small IP packets are predominant in a real network [4]. Moreover, in the current OPS technology, a typical guard time of 50 ns must be inserted between optical packets [5]. This requires that optical packets must be long enough to overcome the resulting link efficiency problems, and hence,

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 765–777, 2006.

a possible issue is the aggregation of several IP packets in a single electronic macro-packet with fixed size, called an aggregate packet which constitutes the payload of an optical packet. Current researches on aggregation have focused on the filling ratio of the optical payload and on the aggregation delay (e.g., our previous works [6], and [7]), without giving the impact of such aggregation on the overall network performance and design. Furthermore, IP packets with same CoS and destination are aggregated together which means that the filling ratio of the aggregate packet risks being very low if the number of destinations becomes large. The two aggregation techniques (DAT and WATT) are suggested to overcome this limitation by aggregating IP packets regardless of their destinations. The proposed application of the aggregation methods relies on a slotted version of the metropolitan area network architecture called DBORN (Dual Bus Optical Ring Network) [8], where a high bandwidth efficiency can be achieved. The presence of IP packets with different destinations in one optical packet does not incur an additional processing complexity at intermediate nodes. This is because in slotted DBORN (SDBORN), the optical packet is converted to the electronic domain only after being received by a ring node which extracts IP packets destined for it and locally drops those addressed to other nodes (see section 5).

The present paper is organized as follows. The aggregation techniques are described in section 2. Section 3 presents the analytical model in the case of DAT. In section 4 we present performance comparisons between the two aggregation techniques, while in section 5 we present a network application for our proposals. Finally, section 6 concludes the paper.

2 Description of the Aggregation Techniques

We present the description of DAT under the SP discipline, while WATT is explained under the modified PP discipline (henceforth denoted by PP for simplicity) which can be reduced to the SP one (see [1]). To be fair, all comparisons between WATT and DAT have been carried out under the SP discipline.

2.1 The Deterministic Aggregation Technique (DAT)

Let there be J classes of packets (throughout this paper the term "packet" stands for "IP packet"), where packets with a smaller class number have a higher priority than packets with a larger class number. Each class of packets has its own queue (called arrival queue as shown in Fig. 1) and the buffer of the queue is infinite[1]. A timer with time-out value τ is implemented, and at each timer expiration (i.e. at instants $\{n\tau, n = 0, 1, 2, \ldots\}$) an aggregation cycle is initiated by collecting IP packets from higher priority queues before those of lower priority. Two possibilities of aggregation exist: aggregation without segmentation and aggregation with segmentation. In the first one, if the size of a packet at the head

[1] Our target is to compare the packet delay and bandwidth efficiency under different scenarios (see section 5). For this reason, we suppose an infinite access buffer.

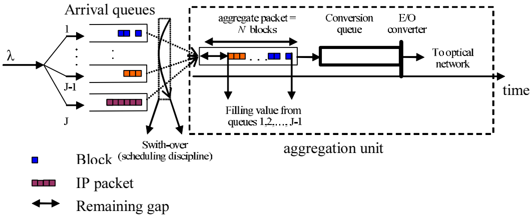


Fig. 1. The aggregation mechanism

of queue i, $\{i = 1, \ldots, J\}$, is greater than the gap, the packet cannot join the aggregation unit. In this case, queues $i+1, \ldots, J$ are checked to serve all packets whose length is smaller than the gap. The aggregate packet will be sent with the existing gap only when queue J is reached (see Fig. 1 where the packet at the head of queue J cannot join the aggregation unit since its length is greater than the gap). In the second one, we allow segmentation of a packet if its size is greater than the remaining gap. That is, in this case the aggregate packet is sent full.

2.2 The Work-Conserving Aggregation Technique with Timer (WATT)

In this case, the aggregation process is governed by the following algorithm.

```
1: Monitor all arrival queues in the system (Fig. 1).
2: Find the set of non-empty queues $NQ$. If all queues are empty, go to Step 1.
3: Launch a timer with time-out value $\tau$.
4: Poll a queue within the set $NQ$ according to the probabilistic algorithm in [1].
5: Fill the aggregate packet from the polled queue. If the aggregate packet
   becomes full, send it to the conversion queue (cancel the running timer) and
   go to Step 1, elsewhere exclude the polled queue from $NQ$ and: go to Step
   4 if $NQ$ is not empty, or to the next step if $NQ$ is empty[2].
6: Wait until at least one empty queue becomes non-empty before the timer
   expiration. In the former case update $NQ$ (ignore excluded queues) and go
   to Step 4, while in the latter case (timer expires) send the aggregate packet
   to the conversion queue and go to Step 1.
```

In the original PP discipline, one and only one packet is served if the system is not idle [2]. However, in our approach the maximum number of IP packets will be served (transmitted to the aggregation unit) when a queue is polled.

Note that we neglect the transmission time of IP packets from the arrival queues to the aggregation unit, which is called the aggregation transmission time (see section 4).

[2] In the case of aggregation without segmentation a polled queue is excluded if it becomes empty or if the packet at its head is greater than the remaining gap. In the case of aggregation with segmentation, a polled queue is excluded only if it becomes empty.

3 Analytical Approach

In [6] we have presented a mathematical model depicting the case of WATT with only one class and one type of packet size distribution. In the following we give an analytical model for the case of DAT with segmentation. Each packet is modelled by a batch of blocks having a fixed size of b bytes (see Fig. 1). Let X be the batch size random variable with probability generating function (PGF) $X(z)$, and probability mass function (pmf) $\{x_n = P(X = n), n \geq 1\}$[3]. The size of the aggregate packet is fixed to N blocks ($N > \max(X)$). We assume two queues and independent arrival Poisson processes (only for the analytical model), with rates λ_1 and λ_2 packet/s. We define $\{A_t^c, c = 0, 1, 2\}$ as the number of blocks arriving at queue c, (for $c = 0$ the queue corresponds to the combination of queues 1 and 2), during an interval of time t, and we denote by $\left\{A_t^c(z) = e^{\lambda_c t(X(z)-1)}\right\}$ its PGF. First, we obtain the probability distribution of number of blocks just before a timer expiration. For this purpose, we choose a set of embedded Markov points as those points in time which are just before timer expirations. Let $t_0, t_1, \ldots, t_n, \ldots,$ be the epochs of timer expirations and define $\{Y^c(t_n), c = 0, 1, 2\}$ by the number of blocks in queue c at instant t_n. Now let $Y_n^c = Y^c(t_n^-)$. Since the whole system (queue 0) and queue 1 behave in a similar way (i.e., at a random epoch, the packet at the head of the queue observes a gap of N blocks), the steady state distribution for $\{Y_n^c, n = 0, 1, 2, ...\}$ is obtained by the same manner for $\{c = 0, 1\}$:

$$y_k^c = \lim_{n \longrightarrow \infty} P(Y_n^c = k), \qquad k \geq 0$$

The following state equation holds for $c = 0, 1$:

$$Y_{n+1}^c = \mid Y_n^c - N \mid^+ + A_\tau^c \tag{1}$$

where $\mid c \mid^+$ denotes $max(0, c)$. The equilibrium queue length distribution (in number of blocks) at an arbitrary time epoch is then described by the probability generating function $Y^c(z)$, which can be derived from (1) in a straightforward and well-known fashion. It is given by:

$$Y^c(z) = \frac{A_\tau^c(z)(z-1)(N - E[A_\tau^c])}{z^N - A_\tau^c(z)} \prod_{k=1}^{N-1} \frac{z - z_k}{1 - z_k} \tag{2}$$

where, $z_1, z_2, \ldots, z_{N-1}$ are the $N-1$ zeros of $z^N - A_\tau^c(z)$ inside the unit circle of the complex plane (there are exactly $N-1$ zeros as proved by the well-known Rouche theorem), and $E[...]$ is the expectation value of the expression between square brackets. The complexity of computation of (2) depends on N. However, even for large N, it is possible to obtain $Y^c(z)$ by resolving $z^N - A_\tau^c(z)$ using MATHEMATICA, and to obtain its corresponding pmf (y_n^c) by using the inverse fast fourier transform (ifft), in a few seconds. Equation (2) allows us to obtain the pmf of the filling value (i.e. the number of blocks in the aggregate packet). Let $F^c, \{c = 0, 1, 2\}$ be the filling value from queue c, and define the filling ratio

[3] $P(X)$ accounts for $Pr(X)$.

random variable by $F_r^c = F^c/N$. If we denote by $\{f_n^c = P(F^c = n), \quad 0 \leq n \leq N\}$ the pmf of F^c, we obtain ($\{y_n^0, n \geq 0\}$ is the pmf of Y^0):

$$f_n^0 = \begin{cases} y_n^0 & 0 \leq n < N-1 \\ 1 - \sum_{i=0}^{N-1} y_i^0 & n = N \end{cases} \tag{3}$$

Equation (3) gives explicitly the pmf of the total filling value of the aggregate packet, and it will be used later in numerical applications to compute the mean filling ratio. To obtain the steady state distribution for $\{Y_n^2, n = 0, 1, 2, ...\}$, the state equation can be written as:

$$Y_{n+1}^2 = \mid Y_n^2 - G \mid^+ + A_\tau^2 \tag{4}$$

where G represents the gap seen by a packet at the head of queue 2. It is given by $G = N - F^1$, and hence, its pmf defined by $\{g_n, n = 0, 1, 2, \ldots, N\}$ can be obtained easily from (3) (by replacing superscript 0 with 1). The PGF of Y^2 is then given by:

$$Y^2(z) = \frac{A_\tau^2(z)(z-1)(N - E[U])}{z^N - U(z)} \prod_{k=1}^{N-1} \frac{z - z_k}{1 - z_k} \tag{5}$$

where U accounts for the random variable defined by: $U = N + A_\tau^2 - G$, and $z_1, z_2, \ldots, z_{N-1}$ are the $N-1$ zeros of $z^N - U(z)$ inside the unit circle.

Now, let us denote by $\{D^c, c = 1, 2\}$ the random variable standing for the aggregation delay of a packet belonging to class c. To obtain D^c analytically, we decomposed it into three parts: 1) the time period elapsed between the arrival instant of the packet to queue c, and the instant when the packet reaches the head of the queue (D_b^c), 2) the delay due to segmentation when the packet cannot be inserted directly into the remaining gap of the aggregate packet (D_s^c), and 3) the aggregation transmission time, i.e. the delay required to transmit IP packets from arrival queues to the aggregation unit. The aggregation transmission time is neglected in the analysis. By using the Little theorem, we can approximate the average of D_b^c by the mean queuing delay of a random block in queue c, that is: $E[D_b^c] = \frac{E[Y^c]}{\lambda_c \times E[X]}$, where Y^c approximates the number of blocks in queue c at a random instant.

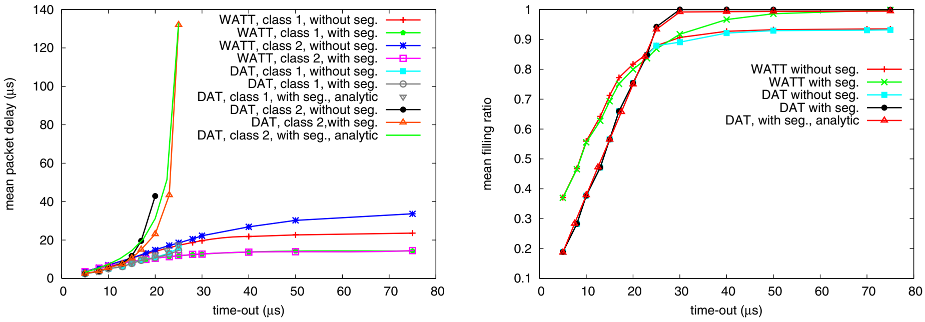
Now, it is easy to show that $D_s^c = \sum_{n=1}^{\infty} p_s^{c,n} \times (n\tau)$, where $p_s^{c,n}$ is the probability that a class c packet is segmented n times before it completely leaves queue c. We give an approximate estimation of D_s^c as follows. First, we suppose that $D_s^{\{c=1,2\}} = D_s^0$. That is, the segmentation delay suffered by a packet belonging to class c is the same as that we obtain if we combine all the CoS queues in one queue. Second, we suppose that $N > \max(X)$, i.e., the aggregate packet size is greater than the maximum size of an IP packet, and hence, the incoming packet is segmented at maximum once (since a packet at the head of queue 0 always finds a gap of N blocks at a random epoch). This leads to restricting our approximation to the first term, i.e., $D_s^0 = p_s^{0,1} \times \tau$. The following is a method to obtain $p_s^{0,1}$: let N_s be the random variable depicting the number of blocks that enter the aggregation unit before the first block of a random packet, given that the latter (the first block of the packet) has entered the aggregation unit (N_s represents the number of blocks having joined the aggregation unit when a part of the packet, i.e. at least one block, joins it). The pmf of N_s, $\{P(N_s = n, n = 0, \ldots, N-1\}$, is given by: $P\left[N_s = n\right] = \sum_{k=0}^{\infty} P\left[K^{0,a} = kN + n\right] = \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} k_i^{0,a} \delta(i - kN - n) = \sum_{i=0}^{\infty} k_i^{0,a} \sum_{k=-\infty}^{\infty} \delta(i - kN - n)$, where $K^{0,a}$ is the number of blocks presented in queue 0 seen at the arrival of the packet, and $\{k_i^{0,a}, i > 0\}$ its pmf. The PASTA property implies that $K^{0,a} = K^0$ (K^0 is the number of blocks at a random instant). In addition, we approximate K^0 by Y^0, i.e. by the number of blocks before a random timer expiration. $\delta(n)$ is the Kronecker delta function, which equals 1 for $n = 0$ and 0 for all other n, and $\{k_i^0 = 0$, for $i < 0\}$. Now we make use of the following identity: $\sum_{k=-\infty}^{\infty} \delta(i - kN - n) = \frac{1}{N} \sum_{s=0}^{N-1} a^{s(i-n)}$, with: $a = e^{j\frac{2\pi}{N}}$. Thus, $P\left[N_s = n, 0 \leq n \leq N-1\right] = \sum_{i=0}^{\infty} k_i^0 \frac{1}{N} \sum_{s=0}^{N-1} a^{s(i-n)} = \frac{1}{N} \sum_{s=0}^{N-1} a^{-sn} K^0(a^s)$, where $K^0(a^s)$ is $K^0(z)$ evaluated at $z = a^s$ (K^0 is always approximated by Y^0). Now it easy to obtain the pmf of N_s by a simple equivalent matrix equation: if P_{N_s} denotes the row vector representing the pmf of N_s, i.e. $P_{N_s} = \left(P(N_s{=}0) \; P(N_s{=}1) \; \ldots \; P(N_s{=}N{-}1) \right)$, and if we define R_{K^0} by the following $(1 \times N)$ matrix: $\left(K^0(a^0) \; K^0(a^1) \; \ldots \; K^0(a^{(N-1)}) \right)$, ($K^0(a^x)$ is $K^0(z)$ evaluated at a^x) we will have:

$$P_{N_s} = \frac{R_{K^1}}{N} \times \begin{pmatrix} a^0 & a^0 & a^0 & \ldots & a^0 \\ a^0 & a^{-1} & a^{-2} & \ldots & a^{-(N-1)} \\ a^0 & a^{-2} & a^{-4} & \ldots & a^{-2(N-1)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a^0 & a^{-(N-1)} & a^{-2(N-1)} & \ldots & a^{-(N-1)^2} \end{pmatrix} \tag{6}$$

where the last matrix in (6) is an $N \times N$ matrix. Now, $p_s^{0,1}$ can be obtained easily by: $p_s^{0,1} = P(X > N - N_s)$, (by using the ifft of $X(z)$ and the pmf of N_s). Note that the analytical model can be extended easily to $J > 2$ classes. In this case, the filling ratio is obtained by combining all queues in one as in (3), and the mean delay of a class i, $2 \leq i \leq J$, is obtained by combining queues 1, …, $i-1$ in one queue with rate $\lambda_1 = \sum_{k=1}^{i-1} \lambda_k$.

4 Performance Comparisons

In this section we focus on comparing the performance of the two packet aggregation mechanisms presented in this paper. The studied parameters are the filling ratio and the aggregation delay (as defined in Section 3). In the sequel, unless mentioned differently, the following assumptions hold. Two classes with equal arrival rates are considered. The arrival process of IP packets is Poisson with an arrival rate $\theta = 900$ Mb/s (the equivalent of λ packet/s in Fig. 1) , and the used IP packet size distribution approximates the real one [4], i.e., 60% of 40-byte packets, 25% of 552-byte packets, and 15% of 1500-byte packets (for the



(a) mean packet delay (b) mean filling ratio

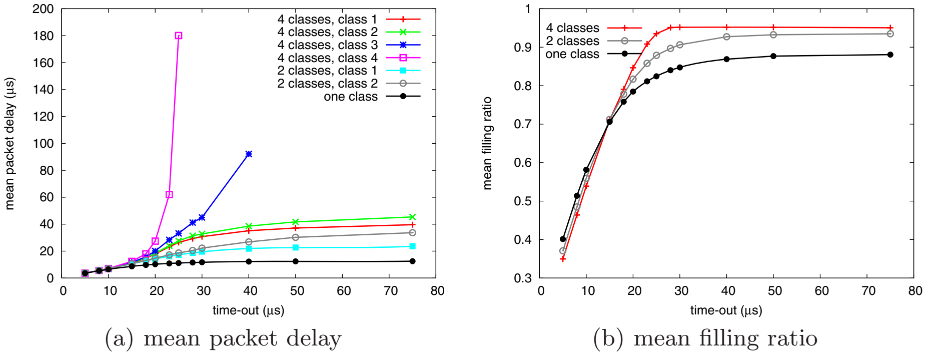
Fig. 2. Comparing the performance of DAT and WATT under Poisson traffic

analytical model we suppose a block size $b = 40$ bytes). The aggregate packet size is 3000 bytes ($N = 75$ blocks for the mathematical model), and the SP discipline is adopted.

Fig. 2 demonstrates that simulation results in the case of DAT with segmentation match very well those of the analytical approach, which proves the accuracy of the mathematical model. Furthermore, the aggregation with segmentation shows better performance than that without segmentation as the time-out increases. This is due to the filling ratio improvement that exhibits the aggregation with segmentation. Moreover, in the case of DAT, a time-out threshold must be respected in order to prevent instability of queue 2 and consecutively instability of the overall system. An upper bound of this threshold can be obtained analytically in the case of DAT with segmentation[4], and by simulation in the case of DAT without segmentation. In Fig. 2, we have observed that a threshold of 22.5 μs in the former case and 20 μs in the latter case are acceptable. Above these values, class 2 packet delay increases abruptly, and the mean filling ratio attains its maximum (1 for DAT with segmentation as Fig. 2(b) shows) since queue 2 is always backlogged. In the case of WATT the system remains always stable regardless of the time-out. This is due to the work-conserving property of WATT and to the negligible value of the aggregation transmission time. This explains also why WATT outperforms DAT when the time-out is smaller than the stability thresholds (DAT allows to deliver empty aggregate packets, which reduces the mean filling ratio as shown in Fig. 2(b)). In the sequel, unless mentioned differently, we consider the case of WATT without segmentation.

Fig. 3 illustrates the impact of service differentiation on the system performance. It can be seen from Fig. 3(b) that if the number of the desired classes increases the filling ratio increases. This is because if a packet belonging to a queue i, has its size greater than the gap, smaller packets belonging to other queues will be allowed to fill the aggregate packet. Fig. 3(a) shows that this benefit affects the packet delay which suffers from an augmentation proportional to

[4] We must have $N > E[A_\tau^0] = \lambda_0 \tau E[X]$ in Eq. 2.



(a) mean packet delay (b) mean filling ratio

Fig. 3. Impact of the presence of several classes with equal arrival rates in the case of WATT

the number of classes. To explain this, take the case when a packet at the head of a queue is greater than the gap. If only one class is available, the aggregate packet is sent immediately to the conversion queue and a new aggregation cycle begins. However, if two or more classes are available, the queue is excluded from the NQ (non-empty queues) set and the aggregate packet is sent only if the timer expires or if all queues become excluded from NQ as explained in section 2.2.

Fig. 4 presents the adavntage of the PP discipline in controlling the level of differentiation between classes (4 classes with equal rates, $\theta = 900$ Mb/s and $\tau = 20\ \mu s$). Each queue is assigned a parameter $0 \le p_i \le 1, i = 1, 2, 3, 4$ as explained in [1]. We consider $p_1 = p_2 = 0.8$, and we modify p_3 from 0.1 to 1 ($p_4 = 1$ as the PP discipline requires, see [1]). The results show that when p_3 increases, the mean packet delay of class 3 is monotonically decreasing and that of class 4 is monotonically increasing, while the delays of the two classes with the highest priorities are almost constant. The filling ratio remains the same under the SP and the PP discipline (this is not shown here).

In Fig. 5 we present the influence of the aggregate packet size and the IP packet size distribution on the aggregation performance ($\theta = 900$ Mb/s, $\tau = 20\ \mu s$ and one class is considered). We consider four packet size distributions

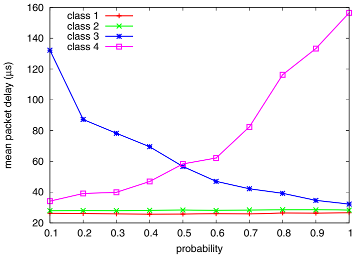
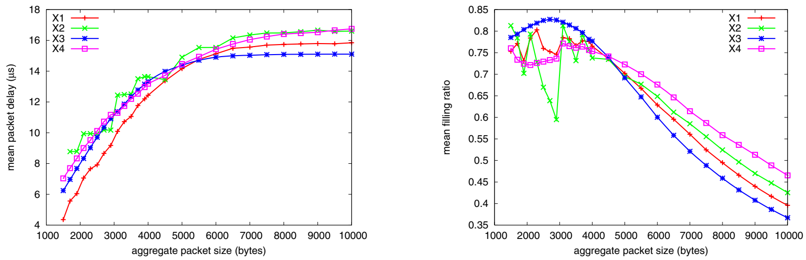


Fig. 4. The effect of the PP discipline



(a) mean packet delay

(b) mean filling ratio

Fig. 5. The impact of the aggregate packet size and the IP packet size distribution

represented by X_1, X_2, X_3 and X_4. X_1 is the distribution that approximates the real one, i.e., 60% of 40-byte packets, 25% of 552-byte packets, and 15% of 1500-byte packets. X_2 is a discrete distribution where big size packets are predominant. That is, we have 60% of 1500-byte packets, 25% of 552-byte packets, and 15% of 40-byte packets. X_3 is exponential with mean $E[X_3] = E[X_1] = 387$ bytes, and X_4 is exponential with mean $E[X_4] = E[X_2] = 1044$ bytes. It can be observed from Fig. 5(b) that the filling ratio fluctuates in different ways for each packet size distribution before it becomes monotonically decreasing. This is because when the aggregate packet size is small, the latter is sent to the conversion queue due to the presence of an IP packet which cannot be inserted into the remaining gap. However, for large values of the aggregate packet size, the latter will be sent due to the timer expiration. This also justifies why the delay (Fig. 5(a)) becomes approximately constant for large values of the aggregate packet size. Note that when big size packets are predominant (i.e., in the case of X_2 and X_4) the mean filling ratio is improved for large values of the aggregate packet size ($>$ 4000 bytes in Fig. 5(b)) at the expense of a little increase in the delay as shown in Fig. 5(a).

The impact of self-similarity is depicted in Fig. 6, where a self-similar traffic with a hurst parameter $H = 0.9$ (which represents a high degree of burstiness) is generated by the method presented in [9]. In the case of WATT, it can be seen that the filling ratio is enhanced under the self-similar traffic with respect to the Poisson one, while the delay remains approximately the same under both types of traffic. This is mainly due to the fact that aggregation cycles are performed successively as long as the queues are not empty and since aggregation transmission time is negligible. However, in the case of DAT, the arrival of bursts leads to abruptly increasing the queueing delay. Furthermore, in this case the mean filling ratio is not meaningfully affected under both types of traffic (self-similar and Poisson). This is because aggregation cycles are performed at fixed instants regardless of the arrival pattern, and hence in the case of self-similar traffic the filling ratio attains the maximum when bursts arrive and the minimum in the absence of burst arrival. In the case of Poisson traffic, the filling ratio remains close to its mean since at high loads, Poisson traffic arrival rate becomes more constant as the packet inter-arrival duration decreases.

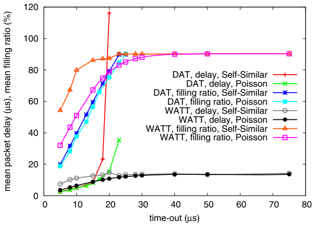


Fig. 6. The effect of self-similar traffic

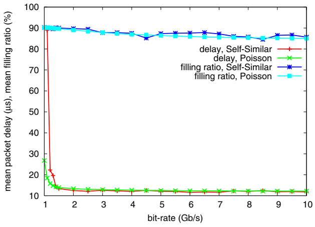
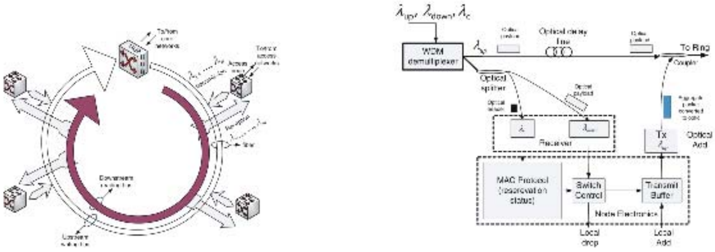


Fig. 7. The effect of the aggregation transmission time

Fig. 7 aims to prove that neglecting the aggregation transmission time is a justifiable assumption in a real system. For this purpose we consider that $\theta = 900$ Mb/s, and we study the impact of the link speed between the arrival queues (i.e., queues $1, 2, \ldots, J$) and the conversion queue (see Fig. 1). It can be observed that a link speed of 1.5 Gb/s gives the same packet delay as that of 10 Gb/s regardless of the arrival pattern. Moreover the mean filling ratio is not affected by the variation of the link speed. This means that a link speed of 1.5 Gb/s is sufficient to consider that the probability of packet (or burst) arrival during the aggregation transmission time is negligible, and hence the latter has no influence on the waiting time of packets and can be neglected in the analysis.

5 Application to a Slotted Network

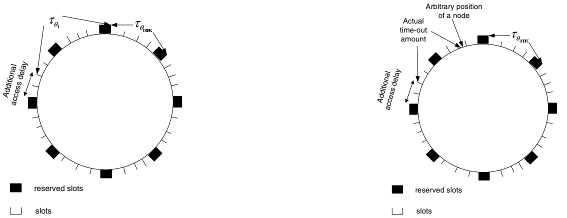
The proposed application is a slotted version of the Dual Bus Optical Ring Network (DBORN) originally proposed in [8]. The topology of the slotted version of DBORN (SDBORN) consists of a ring with two parallel fibers and it is based on separating the transmission and the reception channels through a hub as Fig. 8 shows. The hub delivers empty slots on transmission channels. The upstream channels (λ_{up}) are used for transmission and the downstream ones (λ_{down}) are used for reception. The header of the optical packet is attached to the payload using the out-of-band technique, where headers circulate separately over a control channel (λ_c) (see [5]). We consider that each ring node is equipped with



(a) network architecture

(b) node architecture

Fig. 8. The topology of the slotted version of DBORN (SDBORN)



(a) case of θ variation

(b) case of arbitrary node position

Fig. 9. The additional access delay in two different cases

one fixed transmitter and one fixed receiver, i.e. for each node we assign only one transmission wavelength (λ_{up}) and only one reception wavelength (λ_{down}) as shown in Fig. 8(b). Note that only the control channel is converted to the electrical domain for processing at each ring node, while the bulk of user information remains in the optical domain until it attains the reception channel. This is in perfect conformity with the notion of all-optical (or transparent) networks in literature. In order to ensure a fair access to the ring, we implement a slot reservation mechanism, and we suppose that each node is reserved the same number of slots ($\tilde{n}$ slots) at each ring latency (N_s slots). At the upstream bus, ring nodes detect the header to determine the reservation status of the slot, while at the downstream reading bus, ring nodes preserve the same behavior proposed in the original DBORN, and hence, the optical signal is split, and IP packets are recovered at each node. The latter drops packets which are not destined to it (since aggregation is performed regardless of destinations).

Now let us consider the case of DAT. An aggregate packet (the payload of an optical packet) is delivered to the network each τ as described before. Assume that $\tau_{\theta_{max}}$ is the time-out required (expressed in slots) to warrant a desired level of mean filling ratio (F_{th} %) at the maximum arrival rate θ_{max}. Moreover, we suppose that F_{th} is required regardless of θ variations. Since $\theta < \theta_{max}$, we get $\tau_\theta > \tau_{\theta_{max}}$. Now we assume that each node is reserved $\tilde{n} = N_s/\tau_{\theta_{max}}$ slots[5] (i.e., each node is reserved a slot each $\tau_{\theta_{max}}$). Hence, for a maximum number of ring nodes $M_{max} = \frac{N_s}{\tilde{n}}$, each node finds at each ring latency (N_s slots) as many free slots as timer expirations. This approach guarantees that the delay that an aggregate packet suffers in the conversion queue, before joining the optical network, is at maximum $\tau_{\theta_{max}}$ irrespective of the arrival rate variation and of the position of the node on the ring (see Fig. 9).

In the case of WATT, the number of slots reserved for a node will be given by: $\tilde{n} = N_s/\tau_r$, where τ_r must verify: $\lambda_{max} < 1/\tau_r$ in order to sustain the stability of the conversion which is supposed to be infinite (here, τ_r is different from the aggregation time-out τ, and each node is reserved a slot at each τ_r). λ_{max} is the maximum arrival rate of aggregate packets to the conversion queue. Once τ_r determined, we get the maximum number of ring nodes by the same relation

[5] We suppose that N_s is integer multiple of $\tau_{\theta_{max}}$, since in practice $N_s >> \tau_{\theta_{max}}$.

Table 1. Node number M_{max}, bandwidth efficiency B_w and packet delay

	packet delay (μs)	M_{max}	B_w	τ_r (μs)
No Agg.	296	18	40.5 %	3
WATT	400	30	67.5 %	20
WATT	105	28	63 %	18
DAT	304	30	67.5 %	20 ($\tau_{\theta_{max}}$)

established in the case of DAT, i.e. $M_{max} = \frac{N_s}{\tilde{n}}$, and we obtain by simulations the access delay (in the conversion queue) of an aggregate packet.

In the case of no aggregation, we follow the same reasoning of the case of WATT with the difference that λ will be the arrival rate of IP packets instead of aggregate packets.

Numerical example: we suppose the following assumptions. $\theta_{max} = 900$ Mb/s, self-similar traffic with $H = 0.9$, the IP packet size distribution X_1 (see Section 4), an aggregate packet size $N = 3000$ bytes, $\tau_{\theta_{max}} = 20$ μs, an optical payload of 552 bytes in the case of no aggregation (since the majority of IP packets are small, if the payload is large we get a considerable padding, and if it is small we get a large number of guard times). Also, we considered a channel bit rate $D = 40$ Gb/s, a ring length of 160 km (which corresponds to a ring latency $R_l = 800$ μs), and a guard time of 50 ns. From θ_{max} and the mean of X_1, we obtain for the case of no aggregation, $\lambda_{noAgg} = 0.29 \times 10^6$ packet/s, and hence $\tau_r^{noAgg} < 3.44$ μs (by the stability law of the conversion queue as mentioned before). Now if we consider that the time-out in the case of WATT is 20 μs, we deduce from Fig. 6 that the mean filling ratio (F_{th}) is about 87%, and hence $\lambda_{WATT} = (\theta_{max} \times 10^6)/(F_{th} \times 3000 \times 8) = 0.043 \times 10^6$ packet/s (θ_{max} is also the arrival rate (Mb/s) to the conversion queue since arrival queues are infinite). Thus $\tau_r^{WATT} < 23$ μs. Table 1 shows simulation results under different scenarios. It can be seen that the bandwidth efficiency ($B_w = \theta * M_{max}/D$) is improved when using aggregation (WATT or DAT). This is not surprising since aggregation enhances the filling ratio of an optical payload. The deterministic arrival process of aggregate packets in the case of DAT improves the total packet delay (aggregation delay + access delay in the conversion queue) when compared to WATT where aggregate packets will arrive in bursts in the case of IP self-similar traffic. Hence, although WATT outperforms DAT in terms of aggregation performance (see section 4), DAT is better from an end-to-end view. Note that the packet delay can be also improved at the expense of some loss in B_w by modifying the value of τ_r.

6 Conclusion

We have proposed and analyzed a novel approach for efficiently supporting IP packets in a slotted WDM optical layer with several QoS requirements. Two packet aggregation techniques, called WATT and DAT have been presented and analyzed under Poisson and self-similar traffic. Moreover, an accurate analytical model in the case of DAT with segmentation has been introduced. The results showed that the bandwidth efficiency is improved when using aggregation compared to the standard approach (no aggregation) due to the enhancement in the filling ratio of optical payloads. IP packet delay can be also decreased at the expense of some loss in the bandwidth efficiency.

References

1. Jiang, Y., Tham, C., Ko, C.: A probabilistic priority scheduling discipline for high speed networks. IEEE Workshop on High Performance Switching and Routing. 29-31 May 2001.
2. Jiang, Y., Tham, C., Ko, C.: A probabilistic priority scheduling discipline for multiservice networks. Computer Communications. vol. 25, no. 13, pp. 1243–1254, Aug. 2002.
3. Srivatsa, A., *et al.*: Csma/ca mac protocols for ip-hornet: an ip over wdm metropolitan area ring network. In Proceedings of GLOBECOM'00. vol. 2, San Francisco, CA, 2000, p. 1303-1307.
4. Thompson, K., *et al.*: Wide-area internet traffic patterns and characteristics. IEEE Network. vol. 11, no. 6, pp. 10–23, Nov./Dec. 1997.
5. Dittmann, L., *et al.*: The european ist project david: a viable approach towards optical packet switching. IEEE J. Select. Areas Commun. vol. 21, no. 7, pp. 1026–1040, 2003.
6. Chaitou, M., *et al.*: On aggregation in almost all optical networks. In Second IFIP International Conference on Wireless and Optical Communications Networks WOCN 2005. Dubai,United Arab Emirates UAE, mar 2005.
7. Careglio, D., *et al.*: Optical slot size dimensioning in ip-mpls over ops networks. 7th International Conference on Telecommunications. ConTEL 2003. Zagreb, Croatia, 2003, pp. 759–764.
8. Sauze, N. , *et al.*: A novel, low cost optical packet metropolitan ring architecture. In European Conference on Optical Communication (ECOC 2001), vol. 3.
9. Willinger, W., Taqqu, M.S., Sherman, R., Wilson, D.V.: Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level . IEEE/ACM Trans. Networking. vol. 5, no. 1, pp. 71–86, Feb. 1997.

Issues on Performance Assessment of Optical Burst Switched Networks: Burst Loss Versus Packet Loss Metrics

Nuno M. Garcia[1,2], Przemyslaw Lenkiewicz,
Paulo P. Monteiro[2], and Mário M. Freire[1]

[1] Universidade da Beira Interior, Department of Informatics,
6200-001 Covilhã, Portugal
[2] Siemens SA, Information and Communication, RD1, Research,
2920-093 Amadora, Portugal
{nuno.mgarcia, paulo.monteiro}@siemens.com
przemek.lenkiewicz@gmail.com, mario@di.ubi.pt

Abstract. With the increasing interest in optical burst switching (OBS) networks, the performance assessment of this kind of networks became of particular concern. Recently, some authors suggested that burst loss was not a reliable performance assessment metric for OBS networks. Refuting this claim, this paper presents simulation results obtained for a ring network, using real tributary IPv4 packets as source for the burst assembly. It is shown that burst loss, packet loss and byte loss lead to similar results over a wide range of burst assembly scenarios and network loads, using different resource reservation schemes. Therefore, burst loss is a reliable metric and can be used for evaluation of performance of optical burst switched networks, when realistic burst assembly algorithms are considered over real traffic.

1 Introduction

Burst Switched networks were initially proposed by Amstutz in 1983 [1] as a way to benefit from the statistical multiplexing effect, or as initially described, benefit from "improved bandwidth efficiencies". This concept was later re-introduced in Optical Networks, contributing to the Optical Burst Switching (OBS) Network paradigm, initially proposed by Qiao and Yoo around 1999 [2]. When referring to Optical Burst Switching, three major assembly algorithms are used: *time constrained*; *size constrained*; both time / size constrained, also termed the *hybrid algorithm*. Bursts are created by aggregating packets into a larger data entity, which, after being transmitted, must be disassembled at the end node, and its constituent packets forwarded to their ultimate destination.

Burst switched networks performance is often measured in terms of burst loss or burst drop ratio. Recently, [3] proposed that burst loss was not equal to packet loss and these values vary within the same range. Research activities described in this paper show different results for several assembly scenarios, and particularly, that there exists an equivalence relation between burst loss and packet loss, although the latter is of more interest to the end user than the former.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 778 – 786, 2006.

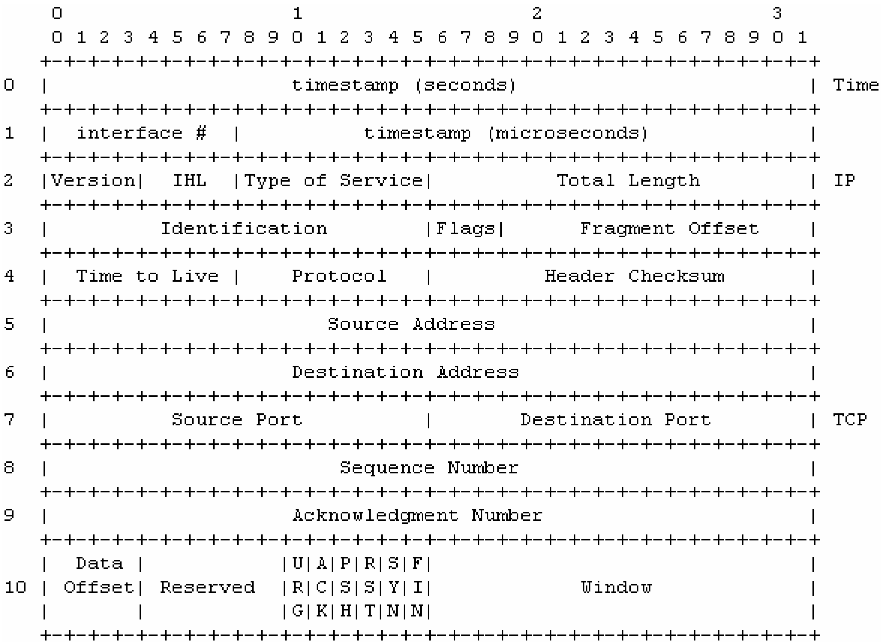
The remainder of this paper is organized as follows: Section 2 discusses basic assumptions and briefly describes the assembly algorithms implemented in the simulator. Section 3 is devoted to the simulation of the burst assembly process. Section 4 discusses the role of burst loss versus packet loss metrics in OBS networks. Section 5 presents main conclusions.

2 Basic Assumptions and Burst Assembly Algorithms

Data packet assembly is a process in which individual data packets are grouped together before the resulting burst is sent into the network structure. These packets may experience re-encapsulation (or not, depending on the network scenario) and typically the nature and origin of the data packets under consideration is not relevant to the assembly principle, as these may be Ethernet frames, ATM cells, IP packets, and so forth. The assembly process requires only the other end of the transmission link to run a complimentary burst disassembly process, retrieving the original constituent packets. In this study, IPv4 packets were used and no encapsulation of the aggregated packets was performed. We can expect IPv6 traffic to output equivalent results, following the research presented in [4]. The disassembly mechanism should thus consider the first 20 bytes of the data burst to be an IPv4 header, and proceed to extract that packet from the aggregated data. This step is repeated until no data is left within the burst. If the network implements burst segmentation techniques, the last readable packet may be corrupt, and if so, it is discarded.

Packet used in the simulation are real IPv4 packets, recorded from NLANR and obtained in [5]. This data is presented in files that record data packet traces in a ***time stamped header*** (*tsh*) format, shown in Fig. 1. The *.tsh* file format stores the payload stripped data packets, time stamped at their acquisition. The typical IPv4 data header is extended by application of the timestamp field (4 bytes for second timestamp and 3 bytes for microsecond timestamp), expressing the timestamp of the captured data packet relative to the 1st of January 1970. The *tsh* record also contains TCP information, comprising Source/Destination ports, Sequence/ Acknowledgment numbers and other TCP specific information. The standard format of the *.tsh* data packet header is shown in Fig. 1.

In order to assure IP address security, the Source and Destination Addresses disclosed in the IP *.tsh* packet header section are hashed to preserve the anonymity of the original machines. However, the IP hashing algorithm [6] is designed in such a manner that it preserves the IP address space density, thus class A servers shall always have lower hashed IP number than class D machines. The source code for the IP address hashing procedure is available from the NLANR website. Packet payload is not recorded. Issues on addresses are important because burst assembly is primarily performed in a “by destination” basis. The simulation handled the computation of the destination addresses for the bursts based on the destination address in the packet *tsh* data as follows: when an address was extracted from the packet, it was looked up in an address table. This address table contains two entries – the first is the IP address itself, the second is the pseudo-address of the destination machine, which is to perform the final disassembly of the burst. If the extracted IP address is not yet present in the address table, then a random pseudo-address is assigned to it as its



```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
0  |                     timestamp (seconds)                       | Time
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
1  |  interface #  |          timestamp (microseconds)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
2  |Version|  IHL  |Type of Service|          Total Length         | IP
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
3  |         Identification        |Flags|      Fragment Offset    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
4  |  Time to Live |    Protocol   |         Header Checksum       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
5  |                       Source Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
6  |                    Destination Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
7  |          Source Port          |       Destination Port        | TCP
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
8  |                        Sequence Number                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
9  |                    Acknowledgment Number                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Data |           |U|A|P|R|S|F|                               |
10 | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
   |       |           |G|K|H|T|N|N|                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 1. Internal format of the .tsh data packet format from NLANR

destination, and this pair was added to the table. This way, the full initial address space was homogeneous and randomly distributed over the available pseudo-addresses of the destination machines. This task is repeated in each node, as a way to closely mimic the hash of the initial IP address space. As an example, while hashed address *12345* processed in node *A* refers to destination machine *X*, it may refer to destination machine *Y* when the same file is processed by node *B*. destination, and this pair was added to the table. This way, the full initial address space was homogeneous and randomly distributed over the available pseudo-addresses of the destination machines. This task is repeated in each node, as a way to closely mimic the hash of the initial IP address space. As an example, while hashed address *12345* processed in node *A* refers to destination machine *X*, it may refer to destination machine *Y* when the same file is processed by node *B*.

The network topology simulated was a four node ring, of nodal degree 2. Shortest path routing was used and full wavelength conversion was assumed for the OBS simulation, using JIT [7] and also the JET [7] signaling protocols. JIT is an immediate reservation protocol and does not perform void filling, and thus every burst is treated independently of its size. On the other hand, JET is burst size sensitive as it performs delayed reservation and attempts void filling, so burst size is important to maximize the efficiency of network resource reservation.

The topology and the remaining default simulation parameters are not relevant for the focus of this research, as a change in these would only alter the performance of the network in terms of burst loss ratios. The simulation was performed with a large set of parameters to allow a wide range of loss ratio values, and thus test the possible correlations of burst and packet loss over the whole counter-domain.

The assembly of packets follows a specific assembly algorithm. Assembly algorithms are constraint driven, and fall into three categories:

1) Maximum Burst Size (MBS)
2) Maximum Time Delay (MTD)
3) Hybrid Assembly (HA)

Other assembly algorithms, like the ones considering classes of services, build upon one of the aforementioned basic types. In this study no CoS (Class of Service) was considered, mainly because the ToS (Type of Service) field in IPv4 packets does not bear reliable information. This limitation could have been overcome by assigning a given packet to a CoS, according to a pre-defined random distribution, but this would not add to the expected conclusions of this research, so no action was taken.

In the MBS assembly algorithm, the incoming data packets are aggregated consecutively into a burst, until its size exceeds the defined threshold. When this occurs, the last data packet overflowing the current burst will start a new one, while the current burst is transmitted into the network structure.

The MTD assembly algorithm was devised to prevent situations where, while using the MBS algorithm, the rate of incoming packets is so low or the arriving packets are so small, that it takes an unacceptable amount of time to fill up a single burst, resulting in excessive transmission delay for the aggregated packets. The MTD algorithm checks for the time difference between the head packet in the burst and the current local time. The burst is sent into the network as soon as that time difference exceeds the maximum delay time defined, independently of the size of the burst and of the number of packets it contains.

If the traffic flow rate is too high or the incoming packets are big, the MTD algorithm may end up aggregating bursts that are too big. In order to prevent such a situation, a HA algorithm was devised. In this assembly scheme, both thresholds – time and size – are considered simultaneously. Incoming packets are aggregated into the burst until either one of the threshold conditions is met. If an incoming packet overflows the burst size threshold, then the burst is close and this packet start a new burst.

3 Burst Assembly Simulation

The algorithm used for burst assembly in this research was HA, with several different thresholds. Thresholds were varied to allow HA to emulate MBS, with time threshold set too high, and MTD, with size threshold is set too low, for current network load. Thresholds used for burst size were set to 64KB and 9 KB, and assembly time varied from 100 μs to 2000 μs for 64, 16, 12, 8, 4 and 1 user in each node. Time thresholds and user load were combined to assure that burst loss really occurred in the network – burst loss ranged from 1.445% to 98.966%.

Burst assembly algorithms using real IP traffic were studied in [8]. Fig. 2 shows how different sets of thresholds change the inter-arrival time between bursts, and consequently define the optimum zone for burst assembly algorithms, defined as corresponding to the minimum interarrival time between bursts with the maximum burst size.

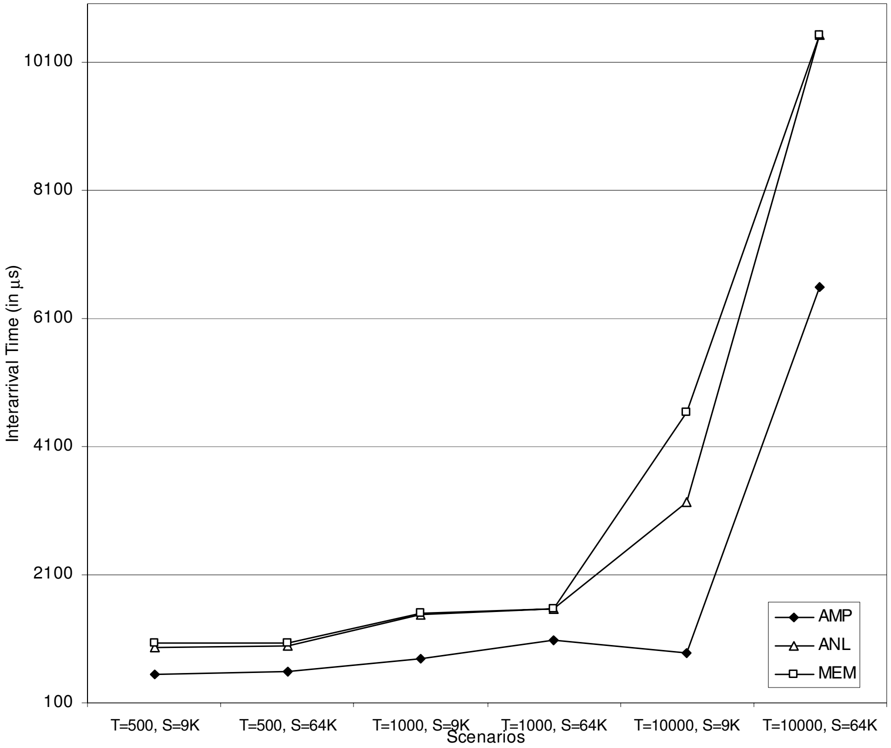


Fig. 2. Burst Inter-arrival time for different threshold scenarios considering three network collection points (AMP = AMPATH, Miami, Florida, USA, ANL = Argonne National Laboratory to STARTAP, MEM = University of Memphis)

Since burst assembly thresholds are network point dependent [8], HA was used with a wide set of thresholds as to obtain a large range of burst characteristics. The result was the creation of bursts very differentiated in terms of Size (in Bytes) and Size (in number of Packets), results that are clearly visible in Fig. 3. The values ranging from 0.905% to 85.043% show the ratio of standard deviation calculated over the averaged Burst Size (in Bytes) and Burst Size (in Packets).

The research relevant results the simulator provided were: Number of bursts, size in bytes for each burst, size in packets for each burst, for both bursts created and bursts dropped. The ratio of – {burst, packets in bursts, bytes in burst} created over dropped was calculated and averaged for several simulations with different simulation time lengths.

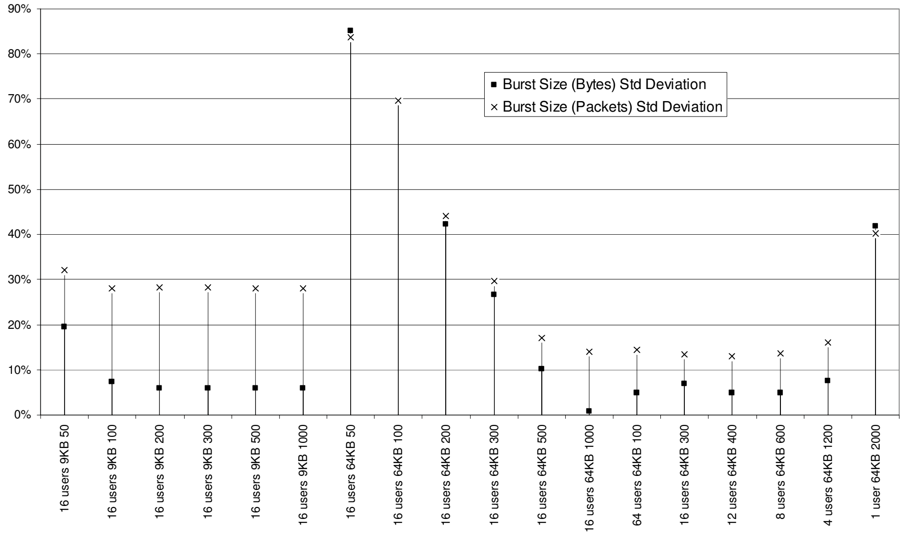


Fig. 3. Standard deviation ratio of average Burst Size (measured in Bytes) and average Burst Size (measured in number of Packets)

4 Burst Loss Versus Packet Loss

The primary metric used for performance assessment of burst switched networks has been burst loss. There are a number of underlying assumptions in this statement that can be expressed in a simplified form, as follows:

1. all bursts are made of independent smaller data entities, which may be called packets (without loss of generalization);
2. all bursts are equally sized;
3. all bursts contain an equal number of packets.

If these three assumptions are hold true, then there is no doubt that burst loss metric is an adequate performance assessment measurement, and what's more, Burst Loss, Packet Loss and Byte Loss ratios are equal. But if bursts are not equally sized, what does it mean that a network lost a burst – exactly how many bytes were in this burst, and what's more, how many packets were lost? That is to say, the Burst Loss metric may not be relevant to real networks, who are know to exhibit self-similar bursty traffic [9-12].

Also, to the end users – machines and humans using the network – burst loss may not be meaningful. The expected network performance and the perceived quality of the service it's supposed to deliver, is measured in terms of "how long and how well is this content taking to travel from machine *A* to machine *B*", and this often means "how many packets were lost" and "how delayed the packet were". This also points out to conclusions already known from the study of burst assembly algorithms using real IP packets: minimum packet delay and maximum burst size, i.e. optimization of burst assembly process, is achieved for the HA algorithm using time and size thresholds that are function of the network load on the burst assembly machine, and thus, are network point dependent [8]. As a result of the optimization of the burst assembly process, it has to be assumed that realistic burst switching deals with bursts that are not homogeneously sized, and of course do not contain a fixed number of packets [8].

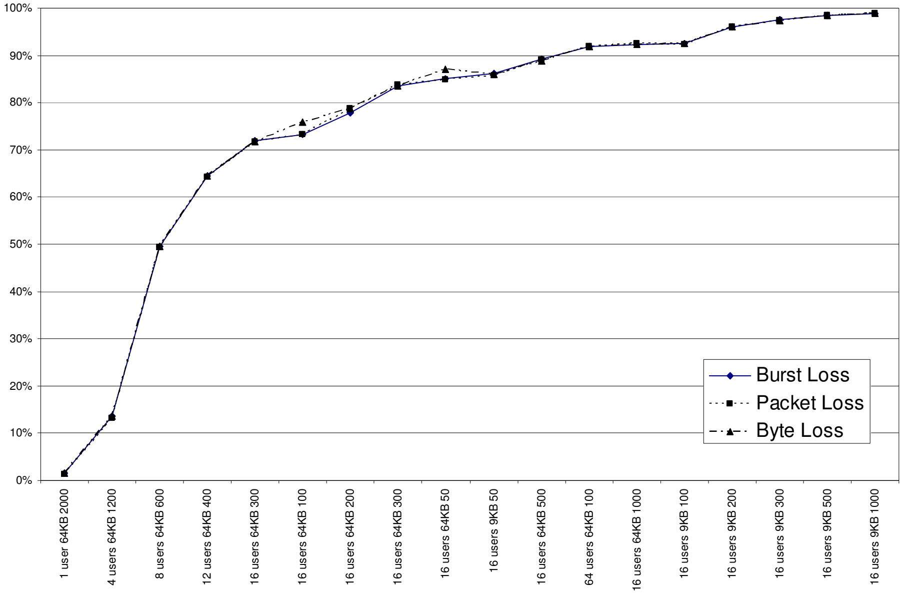


Fig. 4. Burst, Packet and Byte loss for different burst assembly scenarios in an OBS JIT 4-node ring network (time thresholds in x-axis are μs)

If the three above mentioned assumptions can not be held true, as in the case where very heterogeneous burst traffic is generated (the case simulated and presented here), only two alternatives remain: either burst loss is not adequate as a performance assessment metric because it is not equal neither to byte loss neither to packet loss, and the latter would be more “user meaningful”, or with real traffic the simulation proves the Law of Big Numbers, and so, the final results on the network can be assumed as if all the bursts have the same number of packets, and these in turn are equally sized, to the average number of packets per burst the first, and the average number of bytes per packet (and per burst) the latter.

The simulated network was a four-node ring with nodal degree of 2. The network was loaded with bursts assembled from real IPv4 packets, and simulated network data channels were defined as to allow for burst loss.

A set of three ratios was devised and implemented in the simulator:

1. Burst Loss Ratio = number of bursts dropped / number of bursts created at edge nodes;

2. Packet Loss Ratio = sum of the packets in the bursts that were dropped / number of packets assembled in bursts at the edge nodes;
3. Byte Loss Ratio = sum of sizes (in bytes) of bursts that were dropped / sum of sizes (of bytes) of created bursts at the edge nodes.

The three values were calculated for all the simulated scenarios. Fig. 4 and Fig. 5 show the obtained measurements for JIT and JET signalling protocols respectively, with several burst assembly scenarios. As expected, despite such a wide range of burst characteristics in terms of size in bytes and number of constituent packets, and also, despite of the difference in the way the network signaling protocols accepts or drops the bursts, the Burst, Packet and Byte Loss ratios, are almost coincident.

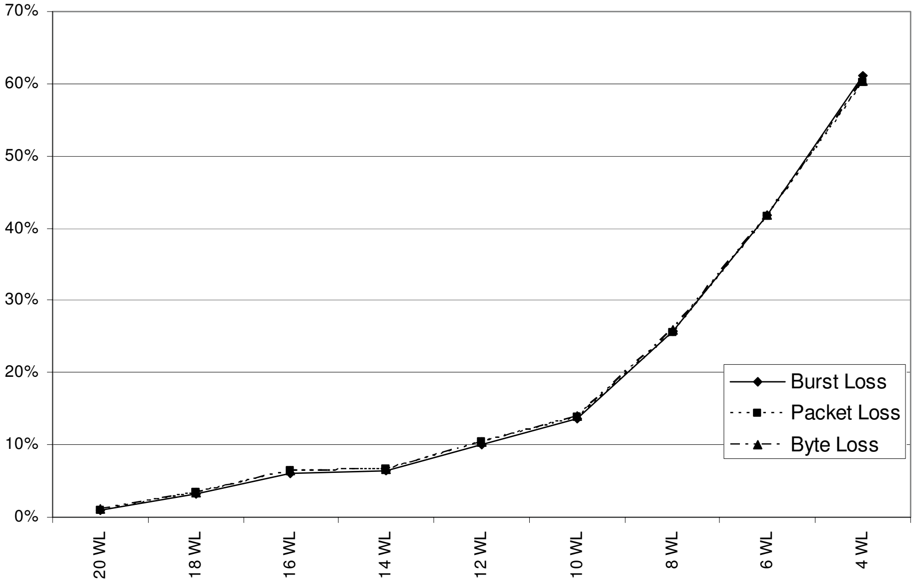


Fig. 5. Burst, Packet and Byte loss for different burst assembly scenarios in an OBS JET 4-node ring network with 64 users, 64 KB burst size threshold, 40 μs time threshold and variable number of data channels (in x-axis of the graph)

5 Conclusion

Kantarci, Oktug and Atmaca [3] have evaluated the issue of burst loss versus packet loss using Pareto distributed traffic generation. When they measured it against Packet Loss for different burst assembly algorithms, their conclusion was that Burst Loss is not a reliable metric for performance assessment of OBS networks, since Packet Loss probability was lower than Burst Loss. On the contrary, results presented in this paper, obtained through simulation using real tributary IP data packets and realistic burst assembly algorithms, show that Burst Loss is a reliable metric for assessment of Burst Switching networks, and that Burst Loss ranges very closely to Packet Loss and to Byte Loss, even when bursts are very heterogeneous in size both packet and byte wise. Also, this study proves that Burst Loss, Packet Loss and Byte Loss are equivalent performance assessment metrics for Burst Switched networks even when the signaling and resource reservation protocols are burst size sensitive, e.g. when void filling is performed (e.g. the JET protocol). Furthermore, it must also be noted that simulation using real tributary data associated with algorithms that are efficiency concerned, produce results that do not always agree with the ones obtained by statistically generated data.

References

[1] S. R. Amstutz, "Burst Switching - An Introduction," in IEEE Communications Magazine, vol., pp. 36-42, 1983.

[2] C. Qiao and M. Yoo, "Optical burst switching (OBS) - A new paradigm for an optical Internet," *Journal of High Speed Networks*, vol. 8, pp. 69-84, 1999.

[3] B. Kantarci, S. Oktug, and T. Atmaca, "Analyzing the Effects of Burst Assembly in Optical Burst Switching under Self-Similar Traffic," in Proc. Advanced Industrial Conference on Telecommunications, Lisbon, Portugal, 2005, IEEE Computer Society Press, pp. 109-114.

[4] N. M. Garcia, M. Hajduczenia, P. Monteiro, H. Silva, and M. Freire, "Modeling and Simulation of IPv6 Traffic," in 7th Internet Global Congress, Global IPv6 Summit, Barcelona, 2005.

[5] National Laboratory for Applied Network Research, "NLANR PMA: Special Traces Archive," in http://pma.nlanr.net/Special/, 2005, accessed at 2005-01-13.

[6] National Laboratory for Applied Network Research, "IPv4 hashing function source code (tsh file format)," in ftp://pma.nlanr.net/pub/dagtools-0.9.6.tar.gz, 2005, accessed at 2005-01-13.

[7] J. Teng and G. N. Rouskas, "A Comparison of the JIT, JET, and Horizon Wavelength Reservation Schemes on A Single OBS Node," in WOBS 2003, Dallas, Texas, 2003.

[8] N. M. Garcia, P. P. Monteiro, and M. M. Freire, "Assessment of Burst Assembly Algorithms using real IPv4 Data Traces," in (submitted) IEEE International Conference on Communications, ICC'06, Istabul, Turkey, 2006.

[9] W. T. Willinger and R. M. S. Sherman, "Self-similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the source level," *IEEE / ACM Transactions on Networking*, pp. 71-86, 1997.

[10] K. Park, "How does TCP generate Pseudo-self-similarity?" in Winter Simulation Conference, 1997, pp. 215-223.

[11] M. S. Borella, S. Uludag, G. B. Brewster, and I. Sidhu, "Self-similarity of Internet Packet Delay," in IEEE ICC '97, 1997, pp. 513-517.

[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (extended version)," *IEEE Transactions on Networking*, vol. 2, pp. 1-15, 1994.

A Multi-hop MAC Forwarding Protocol for Inter-vehicular Communication*

Woosin Lee[1], Hyukjoon Lee[1], Hyun Lee[2], and ChangSub Shin[2]

[1] School of Computer Engineering, Kwangwoon University,
447-1 Wolgye-Dong, Nowon-Gu, Seoul 139-701, Korea
wlee@kw.ac.kr, hlee@daisy.kw.ac.kr
[2] Electronics and Telecommunications Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
{hyunlee, shincs}@etri.re.kr

Abstract. Conventional topology-based routing protocols such as AODV, DSR and ZRP are not suitable for inter-vehicular communication, where the duration of communication lasts extremely shortly. This paper presents a new inter-vehicular communication protocol called the Multi-hop MAC Forwarding Protocol (MMFP). The MMFP avoids explicit path setup in order to reduce the control overhead associated with it, but instead uses the reachability information towards the destination at each hop. Next-hop nodes are determined on-the-fly by contention based on a priority value. The basic operations of the MMFP are conceptually similar to that of MAC bridges and position-based ad-hoc routing protocols. The MMFP is designed to be integrated with the IEEE 802.11 MAC protocol in order to achieve higher efficiency and accuracy in its time-critical operations. It is shown through simulations that the MMFP outperforms the AODV in a realistic inter-vehicular communication scenario in terms of both the end-to-end delay and packet delivery ratio.

1 Introduction

Inter-vehicular communication based on multi-hop wireless networking is attracting a considerable amount of attention as it can not only extend the coverage of infrastructure-based systems but it can also introduce a new set of services in a robust and cost-efficient manner. In the infrastructure-based systems, the radio coverage of a roadside unit (RSU) can be extended by having a node near the edge of the transmission range forward data to nodes outside the range. Imminent collision warning, rollover warning, work zone warning, platooning, cooperative route planning, and peer-to-peer entertainment are some of the public safety and non-safety related applications that can be enabled by the inter-vehicular communication.

Although there is a large body of work on mobile ad hoc network protocols [1-4], most of them are not suitable for inter-vehicle communication. In general, topology-based unicast routing protocols — proactive, on-demand or hybrid of the two — such as DSDV, DSR and ZRP set up a path between two nodes before they exchange data.

* This work was supported by Grant No. R01-2001-00349 from the Korea Science & Engineering Foundation and Research Grant of Kwangwoon University in 2005.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 787 – 798, 2006.

In inter-vehicular communication scenarios, where network topologies change continuously and abruptly, frequent route updates may be necessary. Route update operations, generally based on message flooding, generate an excessive amount of control message overhead which is one of the main sources of large end-to-end delay. The end-to-end delay is one of the most crucial protocol design parameters in the inter-vehicular communication where the duration of communication may be extremely short. Moreover, the control message overhead may cause a significant media contention when communicating nodes are densely populated as in a crowded urban traffic environment [5]. Therefore, a routing protocol with a minimum amount of control overhead in path discovery is desired in inter-vehicular communication.

Position-based routing protocols can forward packets without path discovery or maintenance operation [6-9]. Forwarding decision at each node is made primarily based on the position of the destination and one-hop neighbor nodes. The position information of the destination node is carried in the packet header so that packets can be forwarded by intermediate nodes in the general direction of the destination node. However, unless a separate channel is available for the location service by which the source node to obtain the position of the destination, the position-based routing protocols can suffer from the overhead of location service that scales with $O\left(1/\sqrt{n}\right)$, where n is the number of nodes [6]. This means the overhead of location service has approximately the same complexity as that of path discovery. Furthermore, the inaccuracy of position information caused by node mobility may lead to a significant decrease in terms of packet delivery ratio.

Our goal is to design a new multi-hop routing protocol for inter-vehicular communication that does not perform path discovery or maintenance without using position information. Each node relies on reachability information collected from the packets received previously in making the forwarding decision. This new protocol called MMFP (Multi-hop MAC Forwarding Protocol) is designed as an extension to the IEEE 802.11 MAC layer [10] in order to ensure its functional accuracy in the time-critical operations.

The rest of this paper is organized as follows: In section 2 the MMFP is explained in detail. Simulation results are presented in section 3. Finally, some conclusions are drawn in section 4.

2 Multi-hop MAC Forwarding Protocol

2.1 Main Protocol Operation

The operation of MMFP follows the principle of a MAC bridge that forwards a frame to a particular LAN segment, if the destination address of a frame has been registered to the filter table, and floods it to all LAN segments otherwise. Specifically, whenever a node receives a packet, the addresses of the transmitter, i.e., a 1-hop neighbor, and the source node are entered in the forward table as reachable nodes. Two modes of forwarding are defined: (1) *Implicit unicast mode* is used to select a single forwarding node among the 1-hop neighbors by competition based on a priority value. This mode is used when the reachability information is available for the destination node. (2)

Broadcast mode is used to inform all its 1-hop neighbors to rebroadcast the received packet. This mode is used when the reachability information is not available. A more detailed description on how to maintain the forward table is deferred to the next subsection. The implicit unicast forwarding process is different from the conventional unicast forwarding process. Whereas each node forwards packets to the next-hop along the predetermined end-to-end path in the conventional unicast, each node broadcasts packets with the destination address specified in the implicit unicast. By allowing only one of the neighbor nodes receiving the broadcast frame to rebroadcast it, an operation similar to the unicast is achieved. This is in principle similar to the forwarding process of position-based routing.

The rebroadcast node is selected based on a priority value, which is determined by the effectiveness of forwarding by each neighbor node. The effective period of a forward table entry, Received Signal Strength Indicator (RSSI), the hop count, or the interface queue length are a few examples of possible metrics that can be used to determine the priority value. The position-based forwarding is achieved if the distance to the destination node is used as the priority value. The selected node sends an ACK so that the semantics of original IEEE 802.11 MAC is preserved. The black-burst method that allows a node sending the longest jamming signal to reserve the medium is used in order to have the highest-priority neighbor node send an ACK frame. Once the destination node receives a frame, it sends an ACK frame immediately after SIFS without sending the black-burst signal. If there are many nodes with the same priority, collisions may occur. The MMFP sends the black-burst signal of a random length once again to resolve the collision. Namely, our black-burst process consists of two black-burst phases; the priority-based first phase and the random backoff-based second phase. A more detailed discussion on the two black-burst phases is presented in section 3.3. The main algorithm of MMFP can be summarized as follows:

```
forward_frame():
    if (new frame is received and destination is another node) then
        lookup forward table;
        if (forward table has destination address) then send_delayed_ack;
            if (send_delayed_ack is successful) then send_implicit_frame;
                    else discard frame;
        else if (frame is flooding frame) then send_flooding_frame;
          else discard frame;
        update forward table;
```

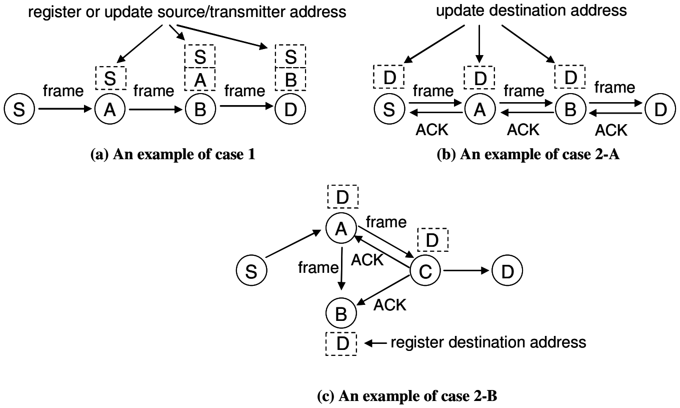
2.2 Maintaining the Forward Table

The main propose of forward table is to provide information about all reachable nodes. Each entry of the forward table consists of two fields (*destination_address, refresh_timer*), of which *destination_address* represents the address of a node reachable and the *refresh_timer* indicates the effective period of an entry. An entry is automatically purged when the value of *refresh_timer* becomes zero.

Depending on the type of frames received, the forwarding table should be updated as follows:

1. *When a data frame is received:* Both the source node and transmitter node are reachable along the reverse path assuming all links are bidirectional. Hence, new entries for the source and transmitter nodes should be registered or the *refresh_timer* should be updated if the corresponding entries exist.
2. *When an ACK frame is received:* There are two sub-cases when an ACK frame is received:
   A. The received ACK frame acknowledges the data frame transmitted by the node itself. The destination node is reachable via a neighbor node. If the transmitted data frame is an implicit unicast frame, it means that the existing entry for the destination node is still valid. Hence, the *refresh_timer* should be reset. Otherwise, a new entry for the destination node should be registered.
   B. The received ACK frame acknowledges the data frame transmitted by a neighbor node. The destination of data frame transmitted by the neighbor node is reachable via the node from which the ACK has been received. Hence a new entry for the destination node should be registered.

Fig. 1 shows an example for each case. In Fig. 1 (c), creation of the implicit multipaths is observed. Implicit multipaths S-A-C-D and S-B-C-D between S and D are created as B adds D to the forward table, and the frame, therefore, can continue to be transferred even if either A or B node moves away. As a result, it is possible to reduce overheads significantly, compared to topology-based routing protocol that is subject to the path maintenance process.


register or update source/transmitter address
S
S
S
A
B
frame
frame
frame
S
A
B
D
update destination address
D
D
D
frame
frame
frame
ACK
ACK
ACK
S
A
B
D
D
A
frame
D
S
frame
ACK
C
D
B
ACK
D
register destination address

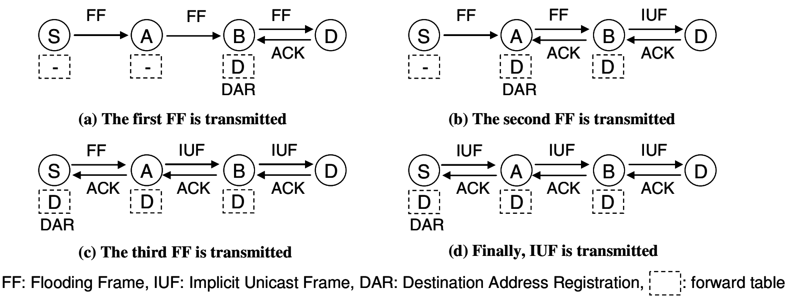
(a) An example of case 1

(b) An example of case 2-A

(c) An example of case 2-B

Fig. 1. An example of forward table maintenance

If the destination is not registered in the forward table, a node should broadcast a flooding frame to all 1-hop neighbors. The flooding frames are repeatedly rebroadcasted by subsequent nodes until they reach a node that has a forward table entry for the destination. From then on the frames are forwarded by the implicit



(a) The first FF is transmitted

(b) The second FF is transmitted

(c) The third FF is transmitted

(d) Finally, IUF is transmitted

FF: Flooding Frame, IUF: Implicit Unicast Frame, DAR: Destination Address Registration, ⬚: forward table

Fig. 2. An example of forward table update process

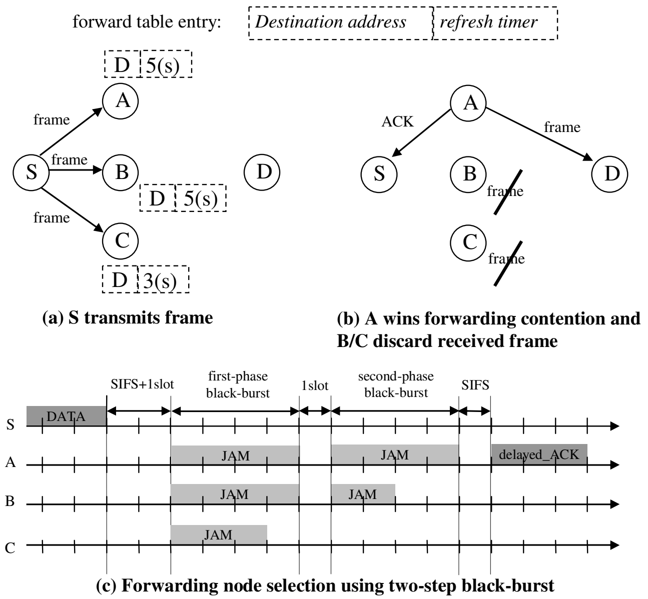
unicast. Since the last nodes that rebroadcast a flooding frame receive an ACK from one of their 1-hop neighbor, i.e., case 2 above, they add a new entry for the destination to their forward tables. This type of forward table update is spread from the destination towards the source as more frames are sent by the same source to the same destination. As a result, the area of flooding is reduced quickly as communication between two nodes proceeds. An example is illustrated in Fig. 2, where none of node A and B initially has a forward table entry for destination node D. The flooding frame sent by node S reaches destination node D via node B. Node D broadcasts an ACK which is received by B. Node B then adds a forward table entry for node D as explained above (Fig. 2 (a)). When node B receives the next frame destined for node D from node A, since node B now has a forward table entry for node D, broadcast an ACK and sends an implicit unicast frame to node D. Upon receiving the ACK from node B, node A adds an entry for node D (Fig. 2 (b)). Similar phases are taken when the next frame is sent by node S and now all of nodes S, A and B have an entry for node D (Fig. 2 (c)), hence no more flooding frames are generated. (Fig. 2 (d)).

2.3 Forwarding Node Selection by Contention

As mentioned previously, all neighbor nodes that have the reachability information for the destination compete for a right to send an ACK using the black-burst method. The winner rebroadcast the frame (i.e., implicit unicast) whereas the losers discard the frame (Fig. 3). This prevents uncontrolled rebroadcasting of the same frame. Since this ACK is delayed by black-burst, we call it a delayed_ACK.

Black-burst method was proposed in [11] and [12] in order to provide guaranteed access delays to rate-limited traffic. By allowing each node transmit a data frame only if the medium is free after sending out an energy burst (channel jamming signal) of which the length is determined independently based on a priority value, a node with the highest priority has the exclusive right to transmit the data frame.

All contending nodes send the black-bursts after they sense the medium is idle in SIFS+1 slot after receiving a data frame. Since it makes no sense to have the destination node contend with other nodes, the destination node is allowed to send an ACK in SIFS after receiving the frame as specified in the IEEE 802.11 standard. In other words, SIFS+1 slot of waiting by the other nodes ensures the priority access to the medium by the destination node taking into account the propagation delay of the ACK.



(a) S transmits frame

(b) A wins forwarding contention and B/C discard received frame

(c) Forwarding node selection using two-step black-burst

Fig. 3. An example of contention-based forwarding node selection using two-phase black-burst

The length of black-burst is determined by:

$$\text{The length of black-burst} = \lfloor (priority\_value) \cdot D_r \rfloor \cdot slot\_time, \tag{1}$$

where *priority_value* is number in [0, 1] that increases as the effectiveness of forwarding by a node increases, D_r is the maximum number of slots allocated to the first phase black-burst, and *slot_time* is the length of a slot (i.e., 9 microseconds).

In our work, we use the value of refresh timer and RSSI in calculating the priority value. The value of refresh timer can be regarded as the validity of reachability information. The RSSI can be used to determine the distance between two communicating nodes based on the path-loss radio propagation model, namely, the ratio of the received signal strength P_{RX} at distance d from the transmitter, to the transmitted signal strength P_{TX}, is given by:

$$\frac{P_{RX}}{P_{TX}} = Cd^{-\alpha}, \tag{2}$$

where C is a constant that depends on the antenna gains, the wavelengths, and the antenna heights, α is the path loss factor ranging from 2 to 4 [13]. Using the distance, the farthest away node from the forwarding node among its contending neighbor nodes becomes the winner. Therefore, it is more likely that the closest nodes to the destination become the intermediate nodes in the forwarding path.

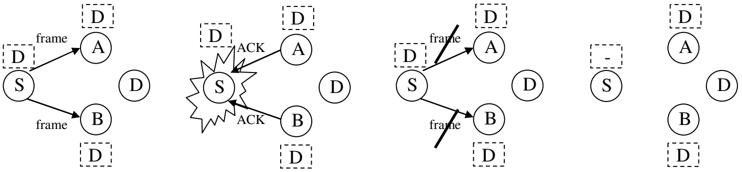
It is possible that more than one contending node have the same priority value and hence the same black-burst length. In this case, ACK's sent by these nodes can collide. In order to resolve the problem of colliding ACK's, all winning nodes perform the second phase black-burst one slot after the first-phase black-burst taking into account the propagation delay of the first-phase black-bursts. The length of the second phase black-burst is determined randomly from the range of allowed slots. Note that the per-hop transmission overhead generated by the two-phase black-burst would not be a significant loss compared to the overhead generated by the transmission of RTS/CTS pair that takes 13 slots in IEEE 802.11 a/g.

In Fig. 3 an example of the selection process of a forwarding node based on two-phase black-burst is illustrated. Three contending nodes (A, B and C) send the first phase black-bursts. In this example, node A and B send the black-bursts of the same length, and node C send a shorter black-burst since node A and B have the same priority values that is higher than node C. In the second-phase black-burst, node A sends a longer black-burst than B as determined randomly. Since A senses the idle channel for SIFS, it proceeds to send a delayed_ACK and rebroadcast the implicit unicast frame, and node B and C discard the frame.

2.4 Maintaining the Sequence Number Table

In the MMFP, the routing loop is prevented by using the sequence number defined in the IEEE 802.11 MAC specification. The sequence number table consists of four fields including *source_address, sequence_number, forwarding_flag* and *refresh_timer.* When a node receives a frame whose source address matches that of a sequence number table entry with a sequence number equal to or smaller than the *sequence_number,* it discards the frame.

The *forwarding_flag* is used to resolve forward table errors due to the collision of delayed_ACK's that may occur because the two-phase black-burst works with a limited number of slots. If two forwarding nodes send the delayed_ACK's at the same time, as shown in Fig. 4, a collision occurs and the sender retransmits the frame for a specified number of times or until it finally receives an ACK. Because the sequence number of all retransmitted frames is the same, the forwarding nodes determine them as duplicate frames and discard them. In this case, the sender, deluding himself that the retransmission has failed, erroneously purges the corresponding entry. The default value of *forwarding_flag* is 0, and it is set to 1 if the frame is forwarded. If the value of *retry field* in the header of duplicated frame and *forwarding_flag* are both 1, the forwarding node recognizes that there has been a collision in sending the previous delayed_ACK, and it retransmits a delayed_ACK.



(a) S transmits a frame (b) ACK collision occurs (C) S retransmits the frame but A/B discard the frame due to the duplicated frame (d) S purges D from the forward table

Fig. 4. An example of forward table error

2.5 An Extension to IEEE 802.11 MAC Protocol

The MMFP uses the four address fields i.e., Address 1 to 4, of IEEE 802.11 MAC frame headers to specify the addresses of the receiver, transmitter, destination and source nodes, respectively. The broadcast address is specified in the receiver address since both the implicit unicast and flooding frames are broadcasted. Because both unicast and broadcast frames are transmitted by using the same broadcast address as the receiver address, the MMFP distinguishes the implicit unicast frames (type: 10, subtype: 1000) and flooding frames (type: 10, subtype: 1001) from each other by using the unused bits of type/subtype fields in the 802.11 MAC header. As opposed to the IEEE 802.11 MAC standard which specifies all broadcast frames are transmitted at the basic rate to minimize the transmission errors of control frames, both the implicit unicast and flooding frames should be transmitted at a data rate.

The MMFP does not use RTS/CTS because all frames are broadcasted, and it resolves the frame loss due to the hidden/exposed terminal problem through retransmission of the unicast frame.

3 Simulation

In order to analyze the performance of MMFP, we performed the simulation using ns-2. The MMFP was implemented in a sublayer between the network and IEEE 802.11 MAC layer. The AODV was also implemented in the sublayer for a fair comparison. We set the values of *active_route_timeout* and *max_rreq_timeout* to 10 seconds, *local_repair_wait_time* to 0.15 seconds, and *rreq_retry* to 3 times as recommended by [14]. The physical layer of IEEE 802.11b was modified to operate as 802.11g by specifying the system parameters for the ERP-OFDM as shown in Table 1. The two-way ground model was chosen as the path-loss radio propagation model. A simulation scenario was designed to reflect the realistic inter-vehicle communication by 180 cars running on a one-way straight-line highway of two lanes

Table 1. Simulation parameters

Parameter	Value
Frequency (GHz)	2.4
CWMin (slots)	15
SlotTime (microseconds)	9
Preamble length (bits)	120
PLCP Header Length (bits)	24
PLCP Data Rate (Mbps)	6
Data rate (Mbps)	54
Transmission range (m)	200
Carrier sensing range (m)	1000
Traffic pattern	CBR
UDP payload size (bytes)	1024

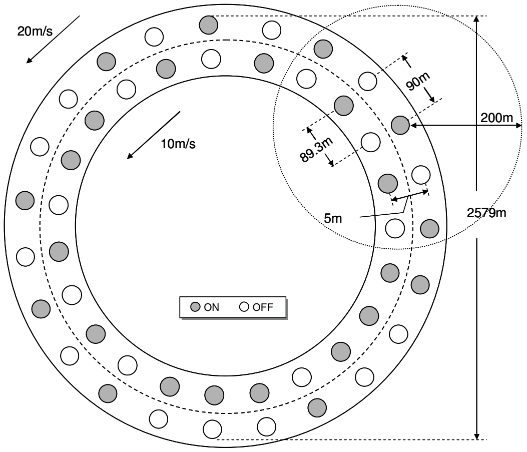


Fig. 5. Circular scenario

with the occasional occurrences of entrances and exits (Fig. 5). Each node periodically makes random transitions with the probability varied from 0.0 to 0.4 between two states, i.e., 'on' and 'off' states, which represent entering and exiting the highway, respectively. Table 1 lists some of the simulation parameters.

The data rate was set to 54 Mbps with the transmission range of 200 meters. The distance between two nodes in the outer and inner lane was set to 90 and 89.3 meters, respectively. Two adjacent nodes in different lanes were initially separated by 5 meters. All nodes in each lane move at the same speed and the difference in speed between two (passing and driving) lanes is 10 m/s. Scenarios for two cars communicating while moving in opposite directions are left out for further investigation in the future. Each node has nine 1-hop neighbor nodes within its transmission range. Each of the 10 randomly selected nodes sends data traffic at 10 pkts/s for 20 seconds to a destination node that is selected to be a specific distance apart at the beginning of a simulation session. Both the source and destination nodes remain in 'on' state during an entire simulation session. Half of the cars are randomly selected to be initially in 'on' state and the other half in 'off' state such that the network topology changes frequently. A series of simulations were run while changing the values of the distance between the source and destination nodes (360, 720, 1080, 1440, 1800 m) and the on/off probability (0.0, 0.1, 0.2, 0.3, 0.4). Each simulation was repeated 30 times with different seed values for random numbers.

The performance of MMFP was measured with two priority values, based on the refresh timer (MMFP-RT) and RSSI (MMFP-RSSI). Fig. 6 and 7 illustrate the performance of MMFP and AODV in terms of the end-to-end delay and delivery ratio, respectively, against the varying on/off probability values. Here, the distance between the source and destination nodes is fixed at 1440 m. Fig. 6 shows the end-to-end delay of MMFP is consistently lower than that of AODV regardless of the values of on/off probability: 14 ms and 13 ms for the MMFP-RSSI, 50 ms and 48 ms for the MMFP-RT and 128 ms and 238 ms for the AODV when the values of on/off

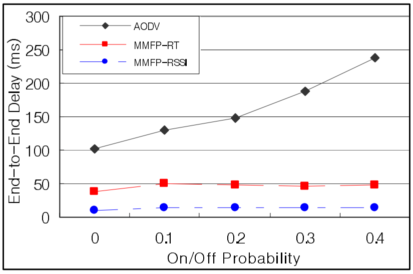


Fig. 6. End-to-end delay vs. on/off probability

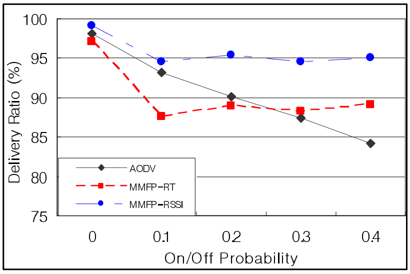


Fig. 7. Delivery ratio vs. on/off probability

probability are 0.1 and 0.4, respectively. We observed the AODV suffer from the frequent local repair of routes which increased the queuing delay and hence the end-to-end delay. By contrast, because the MMFP is able to forward the frames without the route repair via the implicit multi-paths, the end-to-end delay remains almost constant. In particular, the MMFP-RSSI outperforms the MMFP-RT in terms of the mean number of hops from the source node to the destination node, for example, 9.1 hops versus 10.2 hops with on/off probability 0.4. Furthermore, a smaller number of ties in priority values among the contending neighbor nodes occur when RSSI is used to calculate the length of the black-bursts.

In Fig. 7, we can see that the MMFP-RSSI outperform both the MMFP-RT and AODV. However, the AODV achieves a higher delivery ratio (93 %) than the MMFP-RT (88 %) when the on/off probability is 0.1. This is because the MMFP-RT loses more frames due to the reset of queue as well as the hidden terminal problem when a node switches to the 'off' state from the 'on' state than the AODV. However, the amount of frame losses due to the collision decreases quickly enough for both the MMFP-RT and MMFP-RSSI that they outperform the AODV (89% and 95% versus 84% when the on/off probability is 0.4).

Fig. 8 and 9 show the performance of MMFP and AODV in terms of the end-to-end delay and delivery ratio, respectively, against the different values of distance between the source and destination nodes with the fixed value of on/off probability (0.3). In Fig. 8, it is shown the end-to-end delay of MMFP-RT and MMFP-RSSI is lower than that of AODV in all regions of the distance values except at 360 m (2.2 ms for the MMFP-RSSI, 13.5 ms for the MMFP-RT and 12.8 ms for the AODV). When the distance increases to 1800 m, the end-to-end delay becomes 16.9 ms for the MMFP-RSSI, 53 ms for the MMFP-RT and 302 ms for the AODV. The steep increase in the end-to-end delay of AODV is due to the increase in queuing delay caused by the route repairs as the probability of route failure increases with the distance. By contrast, for the MMFP, the end-to-end delay increases slowly as the queuing delay is barely affected by the increased distance. Again, the MMFP-RSSI outperforms both the MMFP-RT and AODV. As shown in Fig. 9, the delivery ratios of MMFP and AODV both drops as the communication distance increase: from 99% to 91% for the MMFP-RSSI, from 98 % to 84 % for the MMFP-RT and from 95 % to 83 % for the AODV, respectively, as the distance increase from 360 m to 1800 m.

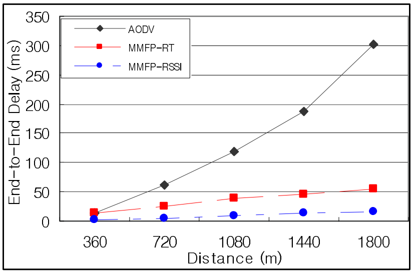


Fig. 8. End-to-end delay vs. intervehicular distance

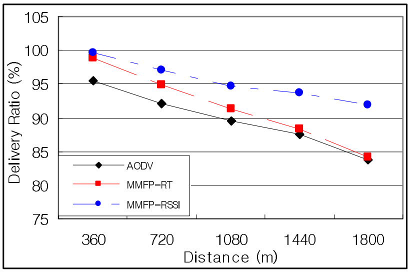


Fig. 9. Delivery ratio vs. intervehicular distance

4 Conclusions

In this paper, we propose a new multi-hop routing protocol for inter-vehicular communication. The proposed protocol, MMFP, does not perform path discovery or use the position information of communicating nodes. Since no path discovery or maintenance is performed, the communicating nodes experience shorter delay which is critical in the high-mobility scenarios of inter-vehicular communication. The fact that the MMFP is implemented as an extension to IEEE 802.11 MAC is a significant advantage in terms of reliable performance and rapid deployment. Additional simulations are being set out to evaluate the performance of MMFP in more realistic situations such as a two-way highway with multiple lanes in each direction and a blind intersection. Further investigations are also underway to improve the performance of the MMFP by integrating position information into the forward node selection procedure and by containing flooding frames within the general direction of the destination node.

References

1. Perkins, C.E., Bhagwat, P.: Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, In Proc. of ACM SIGCOMM'94 (1994) 234–244
2. Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks in Mobile Computing, Imielinski, T. and Korth, H. Eds. Norwell, MA: Kluwer Ch. 5. (1996) 153-181
3. Perkins, C., Royer, E.: Ad-hoc On-Demand Distance Vector Routing, In IEEE Workshop on Mobile Computing Systems and Applications (1999) 90-100
4. Perlman, M.R., Haas, Z. J.: Determining the optimal configuration for the zone routing protocol, IEEE Journal on Selected Areas in Communications (1999) 1395–1414
5. Zhu, J., Roy, S.: MAC for Dedicated Short Range Communications in Intelligent Transport System, IEEE Communications Magazine (2003) 61-67
6. Mauve, M., Widmer, J., Hartenstein, H.: A survey on position-based routing in mobile ad hoc networks, IEEE Network, Vol. 15 No. 6. (2001)
7. Karp, B., Kung, H. T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, MobiCom '00, Boston Massachusetts (2000) 243–254

8. Basagni, S. *et al.*: A Distance Routing Effect Algorithm for Mobility (DREAM), MOBICOM '98, Dallas TX USA (1998) 76−84
9. Blazevic, L. *et al.*: Self-organization in mobile ad-hoc networks: the approach of terminodes, IEEE Communication Magazine (2001)
10. ANSI/IEEE: 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1999)
11. Sobrinho, J.L., Krishnakumar, A.S.: Distributed multiple access procedures to provide voice communications over IEEE 802.11 wireless networks, GLOBECOM'96 Communications: The Key to Global Prosperity, Vol. 3. (1996) 1689 – 1694
12. Jacob, L., Xiang, Li, Luying, Zhou: A MAC protocol with QoS guarantees for real-time traffics in wireless LANs, ICICS-PCM 2003, Vol. 3. (2003) 1962 – 1966
13. Rappaport, T. S.: Wireless communications. principles and practice., Prentice Hall (1996)
14. The Network Simulator(ns-2), http://www.isi.edu/nsnam/ns/

Route Lifetime Based Optimal Hop Selection in VANETs on Highway: An Analytical Viewpoint

Dinesh Kumar, Arzad A. Kherani, and Eitan Altman

INRIA B.P. 93, 2004 Route des Lucioles, 06902 Sophia Antipolis Cedex, France
{dkumar, alam, altman}@sophia.inria.fr

Abstract. We consider the problem of optimal next-hop selection in a route between two vehicles, for a simple scenario of Vehicular ad hoc networks (VANETs) on a highway. For a given approximation of the optimal number of hops, we seek the optimal choice of next-hop based on its speed and inter-node distances, so as to maximize the expected route lifetime. Under a Markovian assumption on the process of speed of nodes, we show that the optimal choice of speeds attempts to equalize the lifetimes of adjacent links. A monotone variation property of the speed of relay nodes under the optimal policy is proved. These properties have been confirmed with simulations. The optimal policies and their structures can assist in enhancing the performance of existing VANET routing protocols.

Keywords: Vehicular ad hoc networks, optimal routing, link lifetime.

1 Introduction

Vehicular Ad Hoc Networks (VANETs) [1, 2, 3, 4] tend to exhibit a drastically different behavior from the usual mobile ad hoc networks (MANETs) [6]. High speeds of vehicles, mobility constraints on a straight road and driver behavior are some factors due to which VANETs possess very different characteristics from the typical MANET models. Broadly speaking, four such characteristics are rapid topology changes, frequent fragmentation of the network, small effective network diameter and limited temporal and functional redundancy [6]. Due to this fundamental behavioral difference between MANETs and VANETs, topology-based routing protocols developed for the former cannot be directly used in the latter. Topology-based protocols are the table-driven *proactive* protocols and on-demand *reactive* protocols [7]. For example authors in [10] have shown that TORA (an on-demand protocol) is completely unsuitable for VANETs. Instead, position-based routing protocols such as LAR, DREAM or GPSR [11, 12, 13] that require a-priori knowledge of vehicles' geographic location (from a GPS service) could be used for VANETs for faster route discovery and improved performance. But position-based routing protocols suffer from geographic routing failures due to presence of *topology holes* [14] and authors in [14] propose *spatially aware routing* for VANETs to overcome this drawback. However optimality of spatially aware routing has not been proved and it could be further enhanced in order to improve performance.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 799–814, 2006.

A routing protocol usually has three main functions: route discovery, optimal route selection (among various candidate routes discovered) and route maintenance. Once an optimal route from a source to its destination has been discovered and selected, route maintenance must be carried out, in order to track link failures (due to movement of relay nodes) and perform route re-discovery. Route maintenance and re-discovery are expensive in signalling and computation, and hence it is desirable to choose the optimal route comprising links with maximum possible lifetimes during the optimal route selection phase. In this paper we propose an optimal route selection criteria from an analytical viewpoint, for the simple scenario of a VANET on a straight line highway. Our optimal route selection criteria consists of the optimal choice of next-hop, based on *maximum route lifetime*. The proposed optimal next-hop selection criteria based on maximum route lifetime is not a competitor to the optimal route selection methods in any of the existing routing protocols, but rather it can be used in conjunction with them. Our goal is not to propose a complete routing protocol along with its implementation aspects. We rather focus to gain an analytical insight into the route lifetime dynamics of a VANET by considering only a simple scenario and our observations on the structural characteristics of the optimal policies can assist in enhancing the performance of any of the existing routing protocols mentioned before or spatially aware routing in particular, for VANETs. As discussed before, optimal route selection in VANETs can be very different from that in MANETs and designing a routing protocol for VANETs can be very complex due to the rapidly changing topology and frequent link breakdowns. In our model, we introduce certain simplifying assumptions, as compared to a real life scenario, in order to gain an analytical insight into the dynamics of vehicle mobility and route lifetimes in VANETs. Without these simplifying assumptions it can be very hard to study these dynamics. For instance, a VANET in city traffic scenario can be very hard to model and our analysis does not hold good for this case. The contributions of this paper are twofold. Firstly, the heuristics and structural characteristics of the optimal hop selection policies developed in this paper can assist in better understanding the dynamics of route lifetime in VANETs. Secondly, the results can serve in enhancing the performance of existing routing protocols for VANETs.

2 Optimization Parameters

We consider VANETs on a straight line highway in which a vehicle can establish connectivity only with other vehicles traveling in the same direction of its motion. In other words we consider ad hoc networks formed by only those vehicles that are moving on the same side of a high way and not the opposite side. Assume vehicles (nodes) traveling on an infinitely long straight highway with L lanes, moving in the same direction on either side of the highway. Each lane i has an associated speed limit s_i. Assume that in a given lane, the nodes travel with a speed corresponding to the speed limit of that lane. In other words, it is assumed that all nodes move on the highway with a discrete set of speeds which consists of the speed limits of each lane. We follow the convention that $s_1 < s_2 < \ldots < s_L$. When a node transits to an adjacent lane due to driver's natural behavior, it now travels with the speed associated with the new lane. Now consider 2 tagged nodes, a source and a destination moving in any two (possibly same) lanes, traveling in the same direction. At time 0, these nodes are assumed to be distance D apart. If D is large enough then these nodes may not be able to communicate with each other directly. Intermediate relay nodes are required for these two tagged nodes to form a VANET. However more than one options (vehicles in front of transmitting vehicle moving with identical or different speeds in the same lane or adjacent lanes, respectively) for the choice of next hop may be available. How would one decide whether to choose the vehicle in the same lane or in adjacent lanes as the next hop. In this paper we address the problem of coming up with an *optimal* choice of next hop (relay node) such that the associated link lifetime and hence the route lifetime, is maximized. The constraints under which this decision should be made are mentioned in detail in Section 3, but here we emphasize on the fact that making such a decision may not be as simple as it seems at first. An evident reason being that the underlying state space over which the route lifetime has to be optimized is composed of different parameters, each representing as a component parameter of the overall optimization problem. Following are the possible optimization parameters that should be considered and the motivation behind their choice is discussed in a predecessor research report [5] on this work:

1. Optimization over Number of Relay Nodes
2. Optimization over Inter-node Distances
3. Optimization over Speeds of the Intermediate Nodes

In the present work, we assume that nodes (vehicles) are equipped with a GPS receiver and we also assume that the optimal number of relay nodes and the speeds of the source and destination nodes are somehow known in advance. Avoiding relatively large values for number of relay nodes, an optimal choice on number of relay nodes can be fairly approximated from the knowledge of transmission range R and position of source and destination nodes obtained from the GPS receiver. Approximate speeds of source and destination nodes can also be obtained from a GPS service. Given this information, we are interested in obtaining the optimal inter-node distances and optimal speeds of relay nodes that result in a maximum possible route lifetime.

3 System Dynamics and Model

3.1 Dynamics of Individual Nodes

The process of changing speed of any individual node due to lane change on the highway is assumed to be an independent stationary ergodic stochastic process. We are thus also implicitly assuming that the vehicles do not leave the highway. It is assumed as well that the vehicles do not change their direction of motion since we consider VANETs formed by only those vehicles that are traveling on the same side of highway in the same direction. In this paper, we restrict ourselves to the case where the changing speed of any node can be modeled as an irreducible aperiodic Markov process, taking a finite set of constant values $\{s_1, s_2, \ldots, s_L\}$. We assume that a node continues to move in lane i with an associated speed s_i, $1 \leq i \leq L$ for an exponential amount of time before changing its lane, or its speed equivalently. This time is exponentially distributed with rate μ_i and we denote that a node in lane i transits to another lane j with probability $P_{i,j}$ with $P_{i,i} = 0$. Even though our analysis holds good for generic transition probabilities $P_{i,j}$, we assume the following natural structure on node transitions in our highway scenario: from state (or, lane) i, a node can transit only to the states $(i-1) \vee 1$ or $(i+1) \wedge L$. Clearly, from state 1 a node can transit only to state 2 and from state L the only possible transition is to state $L-1$.

3.2 Placement of Nodes

We assume that node spread-out along the highway is dense in the sense that in a sufficiently small neighborhood of any point on a lane we can always find at least one node on the *same lane*. This is like assuming that the transmission range R of a node is significantly large as compared to the distances between two successive nodes in any lane. Most of the results in this paper can be extended to the case where we assume that the existence of a node at any point on a lane is itself a stochastic process. However, since we are more interested in the structural results of optimal distances and speed selections, we will assume that this stochastic process is a constant process, i.e., there is always a node at any given point on any lane. It is also assumed that the width of the lanes on an highway is negligible when compared to the transmission range of mobile nodes along the length of highway. We call this assumption as the *straight line communication* assumption.

3.3 Evolution of Inter-node Distances and Node Connectivity

Consider any two nodes i and j (node j is ahead of node i) moving in any two lanes with both the nodes moving in the same direction. Assume that the two nodes have speeds $v_i(t)$ and $v_j(t)$ respectively at time t. Since the two nodes are moving and also have their speeds changing with time due to lane change, the distance between these nodes will also vary with time. Let us denote the distance of node j from node i (measured in the direction of motion) at time t as $d_{ij}(t)$. Assume that node i is the source of transmissions meant for node j. We say that a direct link or single hop route exists between nodes i and j as long as $0 \leq d_{ij}(t) \leq R$, where R is the maximum possible transmission range of a node i.e. a node can successfully transmit at any range $\leq R$.

The distance between any two adjacent nodes i and $i+1$ (node $i+1$ is ahead of node i) of a route denoted simply by $d_i(t)$, forms a stochastic process that begins with an initial value of $d_i(0) = d_i$ and whose evolution over time, $d_i(t)$, depends on the initial speeds of the two nodes. We assume that two successive nodes i and $i+1$ of a route remain connected *only* until when $d_i(t)$ takes a value outside the interval $[0, R]$ for the first time (see Figure 1). The convention followed is that the link between two successive nodes i and $i+1$ of a route, breaks, if either, *1)* node $i+1$ overtakes node i in the direction of motion and the distance between node i and node $i+1$ exceeds R so that node $i+1$ is outside the maximum transmission range of node i, or, *2)* node i overtakes node $i+1$ in the direction of motion. This convention can be easily relaxed to incorporate the case where the link between node i and $i+1$ breaks only when node i overtakes node $i+1$ by a distance R, in the direction of motion. The results of our analysis will still hold good with this relaxed convention.

In brief, we consider nodes i and j to be connected if node j lies within the maximum transmission range of node i *only* in the direction of motion and not otherwise. Note that since the communication devices mounted in the vehicles operate on car battery which is recharged by the vehicle engine, battery-life of nodes is not an issue in our model. All these simplifying assumptions above and in previous Section 2, have been adopted to avoid a very complex modeling scenario, since the main focus is to get an approximate first glimpse of the underlying dynamics of mobility of nodes and route lifetimes in a VANET.

Assume M relay nodes in a route between the source and its destination, with the source being the 0^{th} node and the destination as the $(M+1)^{th}$ node. Let v_0 and v_{M+1} be the velocities of the source and destination nodes and let D be the distance between them. For a given value of M, let $d_i, 0 \leq i \leq M$ be the distance between node i and node $i+1$. We impose that $\sum_{i=0}^{M} d_i = D$ so that the last hop distance $d_M = D - \sum_{i=0}^{M-1} d_i$. For a non-broken route formed by nodes $0, 1, 2, \ldots, M+1$, we require that $0 \leq d_i \leq R$ and let $v_i, 0 \leq i \leq M+1$ be the velocity of the i^{th} node with v_0 and v_{M+1} known in advance. Note that v_is may take any one of the set of constant values $\{s_1, \ldots, s_L\}$ and there are L^M different possible values that the vector $\underline{v} = (v_1, \ldots, v_M)$ can take.

4 The Problem Formulation

In our model described in the previous section, we assume a dense vehicle traffic scenario on the highway. Due to this assumption multiple candidate routes may exist for choosing an optimal route. If multiple candidate routes are available then we want to choose the route with the *maximum* lifetime. We are given that there are $M+2$ nodes, indexed $0, 1, \ldots, M+1$, constituting a route. Node 0 is the source node and node $M+1$ is the destination node. Now consider any two successive nodes i and $i+1$ in the route, that are distance d apart at time zero. Assume also that at time zero, node i is in lane k and node $i+1$ is in lane l such that $d_i(0) = d$, $v_i(0) = s_k$ and $v_{i+1}(0) = s_l$. Let $T(d, v_i, v_{i+1})$ be the expected time after which the link between these two nodes breaks (see Section 3). We refer to the quantity $T(d, v_i, v_{i+1})$ as the *link lifetime* of the link between the successive nodes i and $i+1$ in a route.

For a route comprised of $M+1$ links, our problem is to find an optimal inter-node distance assignment denoted by $\underline{d}^* = (d_0, \ldots, d_{M-1})$, and an optimal speed assignment, denoted by $\underline{v}^* = (v_1, \ldots, v_M)$, to the M relay nodes such that maximum route lifetime is attained. We thus seek the optimal distance vector $\underline{d}^*$ and speed vector $\underline{v}^*$ such that the *least* of the link lifetimes of the route is maximized. Our optimization problem is therefore the following,

$$\underset{\underline{v},\underline{d}}{\text{Maximize}} \ \underset{i=0..M}{\text{Minimum}} \ T(d_i, v_i, v_{i+1}) \ . \tag{1}$$

Instead of solving the above problem directly, we can also attempt to optimize a different, parameterized, objective function. This objective function will coincide with the original one in Equation 1 when the parameter takes a special value. We state here the following theorem whose proof can be found in [5].

Theorem 1. *The solution of the optimization problem in Equation 1 is identical to that of the optimization problem below as $\alpha \to \infty$.*

$$\underset{\underline{v},\underline{d}}{\mathit{Minimize}} \left[\sum_{j=0}^{M} (T(d_j, v_j, v_{j+1}))^{-\alpha} \right]^{\frac{1}{\alpha}} , \tag{2}$$

In fact, we can say something more about the relation between the two optimization problems of Equation 1 and 2 in the following theorem.

Theorem 2. *There exists a* finite α^* *such that the maximizers of optimization problem of Equation 1 are identical to that of Equation 2* for all values of $\alpha > \alpha^*$.

The proof of this theorem can be referred to in the research report [5]. Theorem 2 ensures that there is no discontinuity in the solution of the optimization problem of Equation 2 with respect to the solution of Equation 1, as $\alpha \to \infty$. Working with the objective function of Equation 2 in fact has an advantage that we can optimize it for some *finite* value of $\alpha > \alpha^*$ and elegantly obtain the solution to the optimization problem of Equation 1.

5 Determining the Expected Lifetimes

Having done with the problem formulation, here we seek to obtain explicit expressions for the link lifetimes, to be able to explicitly define the objective function of either Equation 1 or Equation 2. We study the expected lifetime of the connection between two nodes that are d distance apart at time 0 and have speeds s_i and s_j respectively. We use the notation that a pair of nodes k and l is in state s_{ij} when node k is in lane i with associated speed s_i and node l is in lane j with associated speed s_j. Here onwards, along with $T(d, v_k, v_l)$, we will also use the notation $T(d, s_{ij})$ for the link lifetimes of any two nodes, interchangeably. With some abuse of notation we use the same notation for the state s_{ij} and the relative speeds between the two nodes $s_{ij} \overset{\Delta}{=} s_j - s_i$, interchangeably. Consider a pair of successive nodes forming a link in a route as shown in Figure 1. If the second node is within the range R of the first node then using the *straight line*

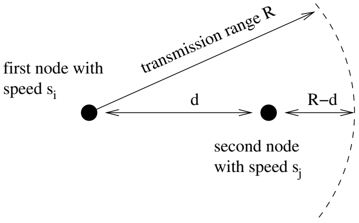


Fig. 1. Two successive nodes constituting a route path

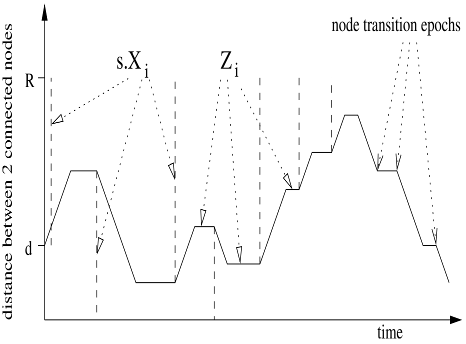


Fig. 2. Random walk model for 2 successive nodes in a route

communication assumption mentioned before in Section 3, the expected remaining link lifetime is given by $T(d, s_{ij})$ and we state the following theorem whose proof can be referred to in [5].

Theorem 3. *$T(d, s_{ij})$ satisfies the following renewal-type recursions*

$$s_{ij} > 0 \quad T(d, s_{ij}) = e^{-(\mu_i+\mu_j)\frac{R-d}{s_{ij}}} \frac{R-d}{s_{ij}} + \int_0^{\frac{R-d}{s_{ij}}} (\mu_i + \mu_j) e^{-(\mu_i+\mu_j)u} \Big[u + \sum_l P_{i,l} \frac{\mu_i}{\mu_i+\mu_j} T(d + s_{ij}u, s_{lj}) + \sum_l P_{j,l} \frac{\mu_j}{\mu_i+\mu_j} T(d + s_{ij}u, s_{il}) \Big] du, \tag{3}$$

$$s_{ij} < 0 \quad T(d, s_{ij}) = e^{-(\mu_i+\mu_j)\frac{d}{|s_{ij}|}} \frac{d}{|s_{ij}|} + \int_0^{\frac{d}{|s_{ij}|}} (\mu_i + \mu_j) e^{-(\mu_i+\mu_j)u} \Big[u + \sum_l P_{i,l} \frac{\mu_i}{\mu_i+\mu_j} T(d - |s_{ij}|u, s_{lj}) + \sum_l P_{j,l} \frac{\mu_j}{\mu_i+\mu_j} T(d - |s_{ij}|u, s_{il}) \Big] du, \tag{4}$$

$$s_{ij} = 0 \quad T(d, s_{ij}) = \int_0^{\infty} (\mu_i + \mu_j) e^{-(\mu_i+\mu_j)u} \Big[u + \sum_l P_{i,l} \frac{\mu_i}{\mu_i+\mu_j} T(d, s_{lj}) + \sum_l P_{j,l} \frac{\mu_j}{\mu_i+\mu_j} T(d, s_{il}) \Big] du. \tag{5}$$

Instead of solving the system of Equations 3, 4, and 5 explicitly in its most general form, we solve it only for some special cases. The main reason for considering only these special cases is that these are the only cases which are of relevance in a real life highway scenario and solutions for cases other than these cannot be applied to real life traffic movement on highways. Another interesting aspect of considering these special cases is that the results that we obtain for these cases constitute a simple form and provide important insights into the structure of the corresponding optimal distance and speed policies. Later with the help of simulations we will attempt to validate the obtained structure for any general case.

In the following sub-sections we attempt to solve the link lifetime recursion equations for particular cases of $L = 2$ and $L \geq 3$. The case $L = 1$ is trivial because there is no breakdown of routes, since all nodes are always traveling with the same speed s_1. Firstly, we consider the case $L = 2$ and, assuming $\mu_1 = \mu_2$, we obtain explicit expressions for the quantities $T(d, s_{ij})$'s. We then solve the optimization problem of Equation 1 directly for $M = 1$ and $\frac{R}{s_{12}} < \frac{1}{2\mu}$. For values of $M > 1$, the global optimization problem can be solved by splitting it into several optimization problems each one of them optimizing over a pair of two adjacent links (i.e., $M = 1$). The solution of these split problems can then be combined to obtain solution to the global optimization problem (for $M > 1$) after taking care of certain coupling issues related to adjacent pairs of links. Second, we consider the case with general values of $L \geq 3$ and $\frac{1}{\mu_i} >> \frac{R}{s_i}$ so that a node remains in lane i for a very long period as compared to the lifetime of a link. For this case we derive only the optimal speed assignment policy, an interesting property of the optimal speed vector solution to the problem of Equation 1 and develop some structural heuristics about the optimal speed vector solution to the problem of Equation 2. Both these cases provide important guidelines on optimally choosing the inter-node distances and speed of next hop.

5.1 $L = 2$

Consider the case where the number of lanes is $L = 2$. There are only two possible speeds s_1 and s_2 in this case with $s_2 > s_1$. At any time t, let the source have speed $v_0(t)$ and destination have speed $v_{M+1}(t)$. Recall that the processes $\{v_0(t)\}$ and $\{v_{M+1}(t)\}$ are assumed to be independent Markov processes over the state space $\{s_1, s_2\}$. The infinitesimal generator matrix is then given by:

$$\begin{array}{c|cc} & s_1 & s_2 \\ \hline s_1 & -\mu_1 & \mu_1 \\ s_2 & \mu_2 & -\mu_2 \end{array}$$

Here μ_i is the rate of the exponentially distributed sojourn time when the process $\{v_0(t)\}$ (or, $\{v_{M+1}(t)\}$) is in state s_i. We state the following lemma without proof.

Lemma 1. *If $\mu_1 = \mu_2 = \mu$ then,*

1. *The process of the speed of destination node with respect to the source node, i.e., $\{v_0(t) - v_{M+1}(t)\}$ forms an irreducible periodic Markov process over (finite) state space $\{0, s_{12}, s_{21}\}$ with the mean sojourn time in any state being exponentially distributed with rate 2μ.*
2. *The state transition probability matrix is of the form*

$$\begin{array}{c|ccc} & s_{12} & 0 & s_{21} \\ \hline s_{12} & 0 & 1 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ s_{21} & 0 & 1 & 0 \end{array}$$

> *In words, from the states with non-zero relative speed, transition is always to the one with a relative speed of* 0 *and from the state with relative speed* 0*, the transition is to either of the other two states, each with probability* 0.5*.*

An important consequence of the observation of Lemma 1 is that the function $T(d, v_i, v_j)$ depends on v_i and v_j only via $v_i - v_j$ with $v_i - v_j \in \{0, s_{12}, s_{21}\}$. We will see later that the observation of Lemma 1 also helps us to compute the function $T(d, 0)$ directly via a simple application of Wald's lemma [8, Chapter 7] without solving any integral equation for $T(d, 0)$. We have the following recursions for $T(d, s_{12})$ and $T(d, s_{21})$ from Equations 3 and 4:

$$T(d, s_{12}) = e^{-2\mu\frac{(R-d)}{s}}\frac{R-d}{s} + \int_{u=0}^{\frac{R-d}{s}} (u + T(d+su, 0))2\mu e^{-2\mu u}du, \text{for} \quad s_{12} > 0, s = s_2 - s_1 \tag{6}$$

$$T(d, s_{21}) = e^{-2\mu\frac{d}{s}}\frac{d}{s} + \int_{u=0}^{\frac{d}{s}} (u + T(d-su, 0))2\mu e^{-2\mu u}du, \text{for } s_{21} < 0, s = s_2 - s_1 \tag{7}$$

For obtaining $T(d, 0)$ we follow the approach of *random walks.* Recall that $T(d, 0)$ is the expected time for which the distance between the two nodes remains in the interval $[0, R]$, starting with distance d apart and 0 relative speed. Clearly, the distance between the nodes can change only when the relative speed between the two nodes is non-zero. The periods of zero and non-zero relative speed alternate and the instants of the beginning of zero relative speed form renewal instants for the relative speed process.

Consider a particle starting at point d. As in random walks, in each time unit the particle moves to either left or right (each with probability $\frac{1}{2}$) and moves by an exponentially distributed amount. The mean of the jump size is $\frac{1}{m}$ where $m = 2\mu$. Let $S_n, n \geq 1$ be the position of particle just after n^{th} jump. It is then seen that $S_n = d + \sum_{i=1}^{n} X_i$ where $|X_i|$s are exponentially distributed random variables (with rate m) corresponding to the jump sizes (see Figure 2). X_i takes negative and positive values with probability $\frac{1}{2}$ each. Let N be the random variable corresponding to the number of jumps required by the particle to exit the interval $[0, R]$ with $R > d$. Let q be the probability that the particle exits via R. The treatment of [8, Chapter 7] can then be used to show that, since $|X_i|$s are independent and identically distributed, $E\sum_{i=1}^{N}|X_i| = E[N]E[|X_1|]$ and $E[(S_N - d)^2] = E[N]E[|X_1|^2]$. To compute $E\sum_{i=1}^{N}|X_i|$, we need $E[N]$ which is derived from the second relation above as follows. Since $|X_i|$ are exponentially distributed, we can invoke the memoryless property of exponential distribution to see that

$$S_N - d = \begin{cases} R - d + Y & w.p.\ q \\ -d - Y & w.p.\ 1 - q \end{cases}, \tag{8}$$

where Y is an exponentially distributed random variable with rate m. Hence, $E[(S_N - d)^2] = E[N]E[|X_1|^2] = qE[(R - d + Y)^2] + (1 - q)E[(d + Y)^2]$

$= (d^2 + E[Y^2] + 2dE[Y]) + q(R - 2d)[R + 2E[Y]]$. From the above expression, since $E[Y] = E[X_1] = \frac{1}{m}$, we can obtain $E[N]$ if we know q. We now obtain q using the fact that $E[S_N - d] = E\sum_{i=1}^{N} X_i = E[N]E[X_1] = 0$ [8]. Now, using the possible values of $S_N - d$ mentioned in Equation 8, $E[S_N - d] = 0 = q(R-d+E[Y])+(1-q)(-d-E[Y])$, hence $q = \frac{d+E[Y]}{R+2E[Y]} = \frac{md+1}{mR+2}$ where we have used the fact that $E[Y] = \frac{1}{m}$. From this value of q, we get (using the fact that $E[Y] = \frac{1}{m}$ and $E[X_1^2] = \frac{2}{m^2}$) $E[N] = ((R-2d)(d+\frac{1}{m})+d^2+\frac{2}{m^2}+2\frac{d}{m})\frac{m^2}{2}$. Assuming $s = 1$ with out loss of generality, it is then seen that $T(d, 0) = E[\sum_{i=1}^{N}(Z_i + |X_i|) - (\sum_{i=1}^{N} X_i - (R-d))I_{\{R-d<\sum\limits_{i=1}^{N} X_i\}} - (-d - \sum_{i=1}^{N} X_i)I_{\{-d>\sum\limits_{i=1}^{N} X_i\}}]$, where Z_is are also exponentially distributed random variables with rate m and they correspond to the time when the distance between the two nodes does not change because of zero relative speed (see Figure 2). Using the memoryless property of exponential distribution, we see that if $I_{\{R-d<\sum\limits_{i=1}^{N} X_i\}} = 1$ then $\sum_{i=1}^{N} X_i - (R-d)$ is (independent and) exponentially distributed with rate m. Similarly, if $I_{\{-d>\sum\limits_{i=1}^{N} X_i\}} = 1$, then $(-d - \sum_{i=1}^{N} X_i)$ is exponentially distributed with rate m. Also, $E[I_{\{R-d<\sum\limits_{i=1}^{N} X_i\}}] = q = 1 - E[I_{\{-d>\sum\limits_{i=1}^{N} X_i\}}]$. Hence, $T(d, 0) = \frac{2E[N]}{m} - \frac{1}{m} = (R-d)md+R+\frac{1}{m}$. We can thus write explicit expressions for the link lifetimes from Equations 6 and 7 as $T(d, 1) = md(R-d) + 2(R-d)$ and $T(d, -1) = md(R-d) + 2d$, respectively.

Optimal Speed Vector Solution to Optimization Problem of Equation 1 for the case of $\frac{R}{s} < \frac{1}{m}$. We consider the case where $\frac{R}{s} < \frac{1}{m}$. This scenario is of relevance since in normal real life highway traffic, a node remains in its lane for an average time greater than the lifetime of the link formed by this node and its next hop. Assuming $s = 1$ with out loss of generality, it is easy to see that for this case $T(d, 1) \leq T(d, 0), \quad d \leq R$, and $T(d, -1) \leq T(d, 0), \quad d \leq R$. Now, let the distance between the source and destination be D such that $R < D < 2R$. Thus one needs at least two hops or equivalently one intermediate relay node for communication. Let the number of intermediate relay nodes be $M = 1$. Also, let the speed of destination with respect to the source be $s = 1$ (i.e. $s_{ij} > 0$). Here we find the optimal speed assignment for a fixed inter-node distance assignment and then later in the next paragraph, we optimize over inter-node distances. So for a given distance d between the source and the intermediate node, the decision is to be made on the speed v of the only intermediate relay node. Let the expected lifetime of the link between source and relay node be $L_1(v)$ and that of the link between relay node and destination be $L_2(v)$. The value of these quantities then are

v	$L_1(v)$	$L_2(v)$
s_1	$T(d, 0)$	$T(D-d, 1)$
s_2	$T(d, 1)$	$T(D-d, 0)$

Now, $T(D-d,0) - T(d,0) = m(D-R)(2d-D)$ and $T(D-d,1) - T(d,1) = (m(D-R)+2)(2d-D)$. Hence, for $d > \frac{D}{2}$, $\arg\max_{v \in \{s_1,s_2\}}(L_1(v) \wedge L_2(v)) = s_1$ and for $d < \frac{D}{2}$, $\arg\max_{v \in \{s_1,s_2\}}(L_1(v) \wedge L_2(v)) = s_2$. Thus, we see that by the solution to the optimization problem of Equation 1, for $s_{ij} > 0$ *the speed of the intermediate node should be the same as the speed of the farther node.* Similarly, it is easy to derive that when the source node has speed s_2 and destination node has speed s_1 (i.e. $s_{ij} < 0$) *the speed of the intermediate node should be the same as the speed of the nearer node.*

Optimal Distance Vector Solution to Optimization Problem of Equation 1 for the case of $\frac{R}{s} < \frac{1}{m}$. As before, let the distance between the source and destination be D such that $R < D < 2R$. Let the number of intermediate relay nodes be $M = 1$ and without loss of generality, let the speed of destination with respect to the source be normalized with $s = 1$. Then for $d > \frac{D}{2}$, it has been shown in the previous paragraph that the optimal speed selection is s_1. Now, it can be shown after simple algebra that for $T(d,0) < T(D-d,1)$ to hold good we must have $d > \frac{D(m(D-R)+2)-R+\frac{1}{m}}{2(m(D-R)+2)}$. Let us denote the RHS of the previous equation by K. Now, if m is such that $K < R$ (and $\frac{D}{2} < K$). Then for $d < K$ we have $min(T(d,0), T(D-d,1)) = T(D-d,1)$. For obtaining optimal d^* we differentiate $T(D-d,1)$ w.r.t. d and equate it to zero, from which we get $d^* = D - \left(\frac{R}{2} - \frac{1}{m}\right)$. For $d > K$ we have $min(T(d,0), T(D-d,1)) = T(d,0)$. For obtaining optimal d^* we differentiate $T(d,0)$ w.r.t. d and equate it to zero to get $d^* = \frac{R}{2}$. For $d < \frac{D}{2}$, it has been shown in the previous paragraph that the optimal speed selection is s_2. It can be shown after simple algebra that for $T(d,1) < T(D-d,0)$ to hold good we must have $d > \frac{m(D-R)D+R-\frac{1}{m}}{2(m(D-R)+2)}$. Denote the RHS of the previous equation by K'. Now, if m is such that $K' > D-R$ (and $K' < \frac{D}{2}$) then for $d > K'$ we have $min(T(d,1), T(D-d,0)) = T(d,1)$. For obtaining optimal d^* we differentiate $T(d,1)$ w.r.t. d and equate it to zero. We thus get $d^* = \frac{R}{2} - \frac{1}{m}$. For $d < K'$ we have $min(T(d,1), T(D-d,0)) = T(D-d,0)$. For obtaining optimal d^* we differentiate $T(D-d,0)$ w.r.t. d and equate it to zero and get $d^* = D - \frac{R}{2}$.

5.2 $L \geq 3$

Some Properties of Solution to Optimization Problem of Equation 2 with $\frac{R}{s_i} << \frac{1}{\mu_i}$. Here we derive some structural properties of the solution to the optimization problem of Equation 2 for the particular case of interest when $\frac{R}{s_i} << \frac{1}{\mu_i}$ so that a node stays in its lane for a time much greater than its link lifetimes. Assume any value of $L \geq 3$ and consider the link lifetime dynamics of two nodes in lanes i and j that are separated by an initial distance $d < R$. It can be easily seen that for $i \neq j$ and $\frac{R}{s_i} << \frac{1}{\mu_i}$, Equations 3 and 4 can be rewritten as $T(d, s_{ij}) = \frac{R-d}{s_{ij}}$ $\forall s_{ij} > 0$ and $T(d, s_{ij}) = \frac{d}{s_{ij}}$ $\forall s_{ij} < 0$. If both the nodes are initially in the same lane, then the distance between these two nodes remains constant till the instant when any one of them changes lanes, so that $\forall s_{ii} = 0$, $T(d, s_{ii}) = \frac{1}{2\mu_i} + \sum_{j \neq i} \frac{P_{i,j}}{2}(T(d, s_{ij}) + T(d, s_{ji}))$. Now consider a route consisting of M intermediate nodes so that the source and destination nodes have speeds v_0 and v_{M+1} respectively, and let the distance vector $\underline{d} = (d_0, \ldots, d_M)$ be fixed. For obtaining the speed vector $\underline{v} = (v_1, \ldots, v_M)$ that maximizes the route lifetime, we can consider minimizing the objective function of Equation 2. Let us make a simplifying assumption here that $T(d, 0) = \infty$ so that $\frac{1}{T(d,0)} = 0$. Though this assumption is not necessary for the analysis that follows, it is well justified here for the case under consideration. We see that the objective function of Equation 2 for any given value of α is given by, $\left[\sum_{j=0}^{M}\left[\frac{1}{T(d_j,v_j,v_{j+1})}\right]^{\alpha}\right]^{\frac{1}{\alpha}}$. Define $f_i(x, y) = \frac{1}{T(d_i,v_i,v_j)}$ such that $x = v_i$ and $y = v_j$. Clearly, if it is allowed to chose an intermediate node i with any arbitrary continuum speed x (thus not restricting to the discrete set of speeds $s_i, 1 \leq i \leq L$), the following condition should be satisfied for an optimal speed assignment to node i, $\frac{d}{dx}\left[(f_{i-1}(v_{i-1}, x))^{\alpha} + (f_i(x, v_{i+1}))^{\alpha}\right]^{\frac{1}{\alpha}} = 0$. This implies, in particular, that $\frac{f_{i-1}(v_{i-1},x)}{f_i(x,v_{i+1})} = \left[-\frac{df_i(x,v_{i+1})}{df_{i-1}(v_{i-1},x)}\right]^{\frac{1}{\alpha-1}}$. Now it is easy to show that $\frac{df_i(x,v_{i+1})}{df_{i-1}(v_{i-1},x)} < 0$. Taking $\alpha \to \infty$, we see that we need $\frac{f_{i-1}(v_{i-1},x)}{f_i(x,v_{i+1})} = 1$, implying that the lifetimes of adjacent links should be *equalized* in order to optimize the objective function of Equation 2. Note that this is only a necessary condition and not a sufficient one, i.e., not all configurations that result in equal lifetimes of adjacent links will be the solution of the optimization problem under consideration. However, *any solution of the optimization problem will satisfy the above mentioned property.* This property also holds good for the case where the speeds of the relay nodes are restricted to a finite discrete set. However, it is obvious that exact equalization of the lifetimes of adjacent links is not achieved due to the lack of the choice of continuum set of speeds for the relay nodes. This issue and another property of *monotone* transition of speeds of relay nodes in an optimal policy has been discussed with detail in [5].

Generic Formula for choice of Optimal Speed of Relay Nodes when $\boldsymbol{\frac{R}{s_i} << \frac{1}{\mu_i}}$. Here we derive a generic formula for the choice of optimal speed of a relay node (solution to the optimization problem of Equation 1) for the particular case of interest when $\frac{R}{s_i} << \frac{1}{\mu_i}$ so that a node stays in its lane for a time much greater than its link lifetimes. Assume any value of $L \geq 3$ and consider the link lifetime dynamics of two nodes in lanes i and j that are separated by an initial distance $d \leq R$. As before, it can be shown that for $i \neq j$ and $\frac{R}{s_i} << \frac{1}{\mu_i}$, Equations 3 and 4 can be rewritten as $T(d, s_{ij}) = \frac{R-d}{s_{ij}}$ $\forall s_{ij} > 0$ and $T(d, s_{ij}) = \frac{d}{s_{ij}}$ $\forall s_{ij} < 0$. Let the speed of source and destination nodes be s_S and s_D and for a 2-hop communication we have $M = 1$. Now, if we assume continuum set of relay node speeds, then for a fixed distance vector, the relay node speeds should be such that the link lifetimes of both links are equal (as seen in the previous paragraph). Therefore if s denotes the continuum speed of the relay node and $R < D < 2R$ then from $\frac{R-d}{s-s_S} = \frac{R-D+d}{s_D-s}$ we get $s = \frac{s_D(R-d)+s_S(R-D+d)}{2R-D}$. This shows that the relay node's optimal speed is a *convex combination* of speeds of source and destination for a two hop route. In particular, at $d = R$ we have $s = s_S$ and at $d = D - R$ we have $s = s_D$. To approximate this continuum speed s with one of the available discrete speeds, we take the following approach. Let s_i be the best approximation to s and let expected lifetimes of the two links be denoted by $L_1(v)$ and $L_2(v)$, where v is speed of relay node.

Case $s < s_i$: If $s < s_i$ then s can either be approximated by s_i or s_{i-1}. For the choice of s_i we have $L_1(s_i) = \frac{R-d}{s_i - s_S}$, $L_2(s_i) = \frac{R-D+d}{s_D - s_i}$ and $L_1(s_i) < L_2(s_i)$. Similarly, we also have $L_1(s_{i-1}) > L_2(s_{i-1})$. Therefore for s_i to satisfy the optimality of Equation 1 we must have $L_1(s_i) > L_2(s_{i-1})$ which results in the following condition on d, $d < \frac{R(s_D - s_{i-1}) + (D-R)(s_i - s_S)}{s_D - s_{i-1} + s_i - s_S}$.
Case $s > s_i$: As in the previous case, with $s > s_i$, s can be approximated by s_{i+1} or s_i. For the choice of s_{i+1} we have $L_1(s_{i+1}) < L_2(s_{i+1})$ and for s_i we have $L_1(s_i) > L_2(s_i)$. Now for s_i to satisfy the optimality of Equation 1 we should have $L_1(s_{i+1}) < L_2(s_i)$ which gives the bound, $d > \frac{R(s_D - s_i) + (D-R)(s_{i+1} - s_S)}{s_D - s_i + s_{i+1} - s_S}$.

Combining the two aforementioned cases and generalizing for any $L \geq 3$, following is a generic formula for the choice of optimal speed of a relay node. Choose s_i as the speed of the intermediate node, if $d \in \left[\frac{R(s_D - s_i) + (D-R)(s_{(i+1)\wedge L} - s_S)}{s_D - s_i + s_{(i+1)\wedge L} - s_S}, \frac{R(s_D - s_{(i-1)\vee 1}) + (D-R)(s_i - s_S)}{s_D - s_{(i-1)\vee 1} + s_i - s_S}\right]$. Note that here $s_S < s_D$ and s_S and s_D can take any values from $s_1, \ldots, s_L$. For $M = 1$, if we assume continuum set of intermediate node speeds as before, then for a fixed distance vector, the intermediate node speeds should be such that the link lifetimes of both links are equal (as seen in the previous paragraph). This implies (it can be shown after some algebra) that the link lifetimes are independent of the choice of inter-node distances, thus implying a *non-unique solution* for the choice of relay node speeds.

6 Simulation Study of a VANET

In order to validate the analysis, we have developed a simulator for a VANET. With this simulator we study and validate only the structural characteristics of the optimal speed assignment policies assuming a fixed inter-node distance assignment. Due to the limitations of this simulator, we do not study the optimal inter-node distance solution. The simulator is based on the model and assumptions proposed in Section 3 and is implemented such that the nodes move in their lanes in a discrete time space. A node in lane i transits to any of the adjacent lanes at the beginning of a time slot of length 0.1 seconds and the transition takes place with probability $1 - p_i$. Given that a node in lane i transits, the transition is to lane j with the same lane transition probability $P_{i,j}$. For our simulations we consider the probabilities $p_1 = \cdots = p_L = p$ to be identical for all the lanes. The probability p is related to μ_i by the relation $\frac{1}{1-p} = \frac{0.1}{\mu_i}$ and for $\frac{R}{s_i} << \frac{1}{\mu_i}$, it is equivalently said that $p \rightarrow 1$. The simulator computes the expected link lifetimes of all possible links by exhaustively simulating over all possible speed assignments $\underline{v}$ of the intermediate nodes for a given scenario of M

intermediate nodes, L lanes, the inter-node distance vector $\underline{d}$, speeds of source and destination v_0 and v_{M+1}, transmission range R, source and destination separation D and the probability p. Once an exhaustive set of link lifetimes for all possible values of $\underline{v}$ is obtained by employing this brut-force method, either of the objective functions of Equation 1 or 2 is applied over this set to obtain an optimal speed assignment policy.

6.1 Simulation Scenarios

A car battery operated mobile device has a typical transmission range of around 200 meters. We therefore consider the possible space of inter-node distances in a VANET to vary from 140 to 200 meters and transmission range of 200 meters is considered for all the simulation scenarios. It has been shown in a previous work [9] that large number of hops in an ad hoc network can significantly degrade the TCP throughput performance. Based on this result, we consider the number of hops $(M+1)$ to vary from 2 to 7 only and the distance between the source and destination nodes is varied from 800 to 1200 meters. We perform simulations for the number of lanes L varying from 2 to 6 and unless explicitly stated in the discussion on the simulation results, the associated speeds are taken as shown in the table that follows,

l	1	2	3	4	5	6
$s_l\ (m/s)$	14	17	22	30	42	55
$\approx s_l\ (km/hr)$	50	60	80	110	150	200

In the following part of this section we discuss some of the scenarios that were simulated and compare their results with the structural results obtained analytically. A more comprehensive simulation study can be found in [5].

1. *Structure of Optimal Policy for $L = 2$ (Section 5.1)*: Figure 3 shows plots of optimal policies obtained from Equation 1 for $L = 2$, $M = 1$, $p = 0.9995$, $D = 300m$ and $\underline{d} = (158, 142)$. The figure clearly illustrates that under optimality, an intermediate node is assigned the speed of the farther node for $s_{ij} > 0$ and that of the nearer node for $s_{ij} < 0$.
2. *Lifetime Equalization over Continuum set of Speeds for $L \geq 3$ and $\frac{R}{s_i} << \frac{1}{\mu_i}$ (Section 5.2)*: In Figure 4 we consider the scenario $L = 3$, $M = 1$, $v_0 = s_3 = 22m/s$, $v_2 = s_1 = 14m/s$, $p = 0.99999$, $D = 300m$ and $\underline{d} = (143, 157)$. In order to be able to validate the equalizing structure obtained in Section 5.2 over a continuum set of intermediate node speeds, we vary the speed associated with lane 2 from $14m/s$ to $30m/s$ in small steps of $1m/s$ and plot the link lifetimes for each such speed of lane 2 separately. This allows the only intermediate node 1 to be assigned one of the quasi-continuum set of speeds for the optimization problem of Equation 2. It is seen in the figure that under optimality, for varying values of v_1, the optimal lifetimes of the links between node 0 and 1 and node 1 and 2 are different. However at $v_1 = 23m/s$ the optimal lifetimes of the two adjacent links are almost equal thus confirming our result obtained in Section 5.2 that

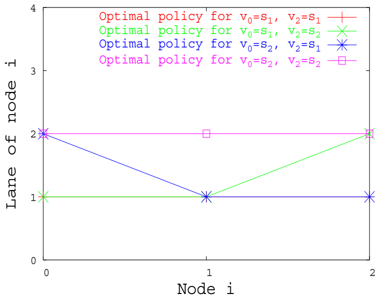


Fig. 3. Structure of optimal policy for $L = 2$

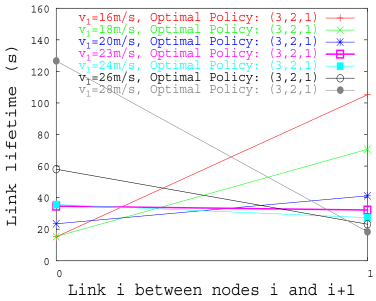


Fig. 4. Lifetime equalization over continuum of speeds

the lifetimes of adjacent links should be equalized in order to optimize the objective function of Equation 2. In fact, it can be observed that we obtain the maximum of the least of the two lifetimes for speed $v_1 = 23m/s$ and the optimal lifetimes obtained for other values of v_1 are not truly optimal because of the unavailability of the choice of speed $23m/s$ in those scenarios.

7 Conclusion

Designing efficient routing protocols for VANETs is quite a challenging task owing to the fast speed of nodes and mobility constraints on the movement of nodes. An attempt has been made in this paper to help accomplish this task better. Under some simplifying assumptions, the analysis of this paper has established that the solution of the optimization problem under consideration tends to equalize the lifetimes of adjacent links in a route. Moreover, there is a monotone variation of the speeds of intermediate relay nodes under the optimal policy. These solution structures have also been confirmed with simulations. The structures obtained are of considerable practical interest as they reduce the space over which an existing VANET routing algorithm would search for the optimal routing policy.

References

1. J. Luo, J.P. Hubaux. “A Survey of Inter-Vehicle Communication”, Tech. Rep., EPFL, Switzerland, ’04.
2. Safe and comfortable driving based upon inter-vehicle communication, “w3.cartalk2000.net”.
3. Car to car communication consortium, “w3.car-to-car.org”. OverDRiVE project, “w3.ist-overdrive.org”.
4. PReVENT: A European program to improve active safety, “w3.prevent-ip.org”.
5. D. Kumar, A. A. Kherani, E. Altman. “Route Lifetime based Interactive Routing in Intervehicle Mobile Ad Hoc Networks”, Research Report, INRIA, France, Sept-2005. w3.inria.fr/rrrt/rr-5691.html.

6. J.J. Blum, A. Eskandarian and L.J. Hoffman. “Challenges of Intervehicle Ad Hoc Networks”, IEEE Transactions on Intelligent Transportation Systems, Vol. 5, No. 4, Dec 2004.
7. Royer et al., “A review of current routing protocols for ad hoc mobile wireless networks”, IEEE Personal Communications, Apr’99.
8. R. G. Gallager. “Discrete Stochastic Processes”, Kluwer, 1998.
9. Gerla et al., “TCP Performance in Wireless Multihop networks”, IEEE WMCSA, Feb’99.
10. S. Jaap, M. Bechler, L. Wolf. “Evaluation of routing protocols for vehicular ad hoc networks in city traffic scenarios”, 11th EUNICE Open European Summer School on Networked Applications, Spain, July’05.
11. Y.B. Ko, N.H. Vaidya. “Location-aided routing in mobile ad hoc netoworks”, MobiCom’98, Oct’98.
12. Basagni et al., “A distance routing effect algorithm for mobility (DREAM)”,‘ MobiCom’98, Oct’98.
13. Karp et al., “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”, MobiCom 2000.
14. Jing Tian, Lu Han, Kurt Rothermel. “Spatially Aware Packet Routing for Mobile Ad Hoc Inter-Vehicle Radio Networks”, IEEE ITSC, Shanghai, China, October 12-15, 2003.

On the Performances of the Routing Protocols in MANET: Classical Versus Self-organized Approaches

Fabrice Theoleyre and Fabrice Valois

CITI, INRIA ARES, INSA Lyon,
21 av Jean Capelle, 69621 Villeurbanne Cedex, France
{fabrice.theoleyre, fabrice.valois}@insa-lyon.fr

Abstract. Mobile Ad Hoc Networks (MANET) are spontaneous wireless networks of mobile nodes without any fixed infrastructure. MANET are promised to a large spectrum of military or civilian utilizations. Routing is a key topic in such networks: overhead must be minimized, optimizing the delay and reducing the packet losses. Several routing protocols were proposed in the literature but, recently, new routing protocols based on a self-organization like Virtual Structure Routing (VSR) were proposed. VSR is based on a self-organized structure with an important stability and persistence. In this paper, we aim to quantify the contribution of the self-organization on the routing behavior and performances. We oppose VSR as a self-organized protocol to the classical one: reactive (AODV), proactive (OLSR) and clustered (CBRP). The impact of the mobility and the density, the horizontal and the vertical scalabilities are studied.

1 Introduction

Mobile Ad Hoc Networks (MANET) are literally networks *ready to work*. All terminals can communicate with other nodes via wireless communications: MANET are spontaneous networks, without any fixed infrastructure. The network must function autonomously, without any human intervention: the self-organization property is vital. In consequence, the nodes must collaborate to set up all network functions fulfilled traditionally by specialized devices. Moreover, a source can be not in the range of the destination: nodes must relay the packets from a source to a destination: the network is multihops. Thus, a distributed routing algorithm must be proposed. Efficient routes must be computed distributively, each node being both router and client. Moreover, all nodes are mobile, creating topology changes. Hence, the network must continuously adapt its knowledge of the topology in order to maintain efficient routes. Ad hoc networks can be connected to the Internet via an Access Point, creating multihops cellular networks.

MANET constitute a wide domain to study. All classical solutions must be re-conceived because of the particular constraints of MANET. The radio links offer a low bandwidth, and create an important instability: fading and multi

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 815–826, 2006.

paths create brutal changes in the radio topology. In the same way, the packet losses are frequent because of collisions. Moreover, MANET are constituted by a collection of different embedded terminal: constraints in energy, CPU and memory are important. Besides, the network is heterogeneous: laptops cohabit with PDA or wired workstations. Thus, propositions must take into account such an heterogeneity, reflecting the different behavior and capacities of the nodes. The load must be balanced efficiently among the nodes, according to their capabilities.

According to us, the self-organization gives multiple answers to the key problems described above. The self-organization is for us the simplification of the topology to facilitate the exploitation of an ad hoc network. The self-organization structures the network, creates a hierarchy, and reduces the topology changes in creating a virtual topology. The self-organization must take into account the network heterogeneity: a node with more power-energy or with a low mobility will contribute more in the network management. In [9], we proposed a virtual topology of self-organization reflecting good properties of stability and persistence. Thus, a routing protocol (Virtual Structure Routing, VSR [11]) benefiting from it was proposed. In this article, we propose to compare a self-organized protocol like VSR with the classical routing protocols existing for MANET. More specifically, one reactive protocol (AODV), one proactive protocol (OLSR) and one hierarchical protocol (CBRP) will be evaluated and simulated. Simulations will measure the performances according to several criteria and environment parameters.

In the first section, we will present a short related work about the self-organization, useful to understand finely the benefits of a self-organization for routing. In the section 3, an overview of the different classes of routing protocols will be given. The section 4 will present the performances evaluation of the different routing classes in MANET. Finally, the section 5 will conclude the article and give some perspectives.

2 Virtual Structures of Self-organization

2.1 Related Work

Backbone. A backbone helps to optimize the traffic exchange, and to reduce the impact of the flooding, minimizing the broadcast storm problem. Only backbone members relay a control packet, reducing the load on the medium.

MANET could be modeled with the graph theory: each terminal is represented by a vertex, and there exists one edge between two vertices if the corresponding terminals have a radio link with each other. In the graph theory, a backbone could be modeled as a Minimum Connected Dominating Set (MCDS): each node is either in the MCDS, or neighbor of at least one vertex of the MCDS. Moreover, the MCDS forms a connected structure of minimal cardinality. MCDS being NP-complete [4], some heuristics must be proposed. A Connected Dominating Set (CDS) is a MCDS without the constraint of minimal cardinality. We extend this notion to a k-CDS: the maximal distance from one node to the CDS is inferior to k hops.

Several articles propose to construct distributively a CDS, minimizing the cardinality [1, 3]. Generally, the algorithms are divided in two major steps: the creation of the dominating set (each node is neighbor of one node of the dominating set), and then its interconnection. A node can be *dominator* (member of the CDS), *dominatee* (the node has a neighbor in the CDS) or *idle*. During the first step, an idle node with the highest weight among its idle neighbors becomes dominator. The degree or the identifier can represent the weight. An idle node neighbor of a dominator becomes dominatee. The second step interconnects the dominators. However, a minimal number of dominators must be chosen. [3] proposes an iterative exploration, starting from the leader, choosing during each iteration to color the locally *best* dominatee. However, such an exploration requires an high delay. [1] proposes a *best-effort* approach: a dominator connects itself to any dominator already connected and at most 2 hops far. The delay is reduced. No maintenance procedure is described although topology changes require to update continuously the structure.

[12] is, to the best of our knowledge, the only localized algorithm constructing a CDS. A node is colored dominatee if the following rule is valid: *no couple of my neighbors are not directly connected*. Else, the node is dominator. This rule forms globally a CDS. The rule was extended further in: *there exists a connected set of neighbors of higher weight which are a dominating set of my whole neighborhood*. This CDS was proposed to optimize the flooding, but not the persistence of the structure. Thus, simulations show that a node changes frequently its role, creating potentially an unstable hierarchy.

Clusters. Clusters allow to divide the network, creating a hierarchy. Clusters constitute natural services areas. The constraint of a cluster can be its diameter: the maximal distance from one node to another node of the same cluster is at most 3 hops. A leader called clusterhead can also be elected and maintained in the cluster. In this case, the radius represents the cluster constraint.

[7] presents an algorithm widely used in the literature to form clusters. Each node initiates a neighborhood discovering. A node with the highest weight among its neighbors without clusterhead is elected clusterhead. Its neighbors join its cluster. During the maintenance, the clusterhead is no longer maintained. When a node detects that the diameter constraint is violated, the nodes decide distributively how to split the cluster. The knowledge of the 2-neighborhood is required. If a clusterhead is maintained, the maintenance can use the radius constraint.

[2] proposes the creation of k-clusters: each node is at most k hops far from its clusterhead. The algorithm could be divided in two waves: the first wave propagates the highest ids, and the second wave propagates the lowest ids. However, no maintenance procedure is described.

2.2 The Virtual Structure Used by VSR

[9] presents a virtual structure combining both a backbone and clusters. The structure is fully integrated in order to reduce the overhead: the algorithms to maintain the backbone and the clusters share some information. This virtual structure is used by VSR [11] to provide a new self-organized routing protocol.

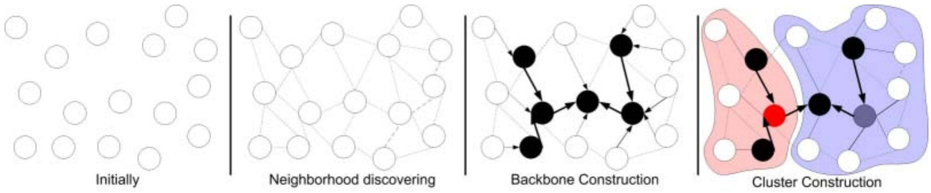


Fig. 1. Virtual structure construction

First, a k_{cds}-neighborhood discovering is triggered: all the nodes exchange periodically `hello` packets to discover all the nodes at most k_{cds} hops far, their identity, weight, state... Then, the algorithm constructs a k_{cds}-CDS in electing dominators and interconnecting them, forming a backbone: each node is at most k hops far from the backbone, k_{cds} being a parameter of the protocol. The backbone construction is triggered by one or several leaders. In an hybrid network, the gateway to the Internet should act as leader. Else, a leader must elected distributively. Finally when the k_{cds}-CDS is locally constructed and based on the sub-graph of dominators, clusterheads are elected such than $k_{cluster}$-clusters are built.

However, the radio topology changes continuously in MANET. Thus, [9] details event-driven procedures to maintain an efficient virtual structure. All the nodes in the network maintain a parent in the backbone, so that each node is at most k hops far from the backbone. Besides, procedures allow to verify distributively the backbone connectivity, and eventually to reconnect it. In the same way, distributed procedures allow to maintain the dominance property of the clusters.

This virtual structure of self-organization was proved to present interesting properties of self-stabilization [10]. Starting from any initial state, potentially corrupted, the algorithms converge to a valid state in a finite time. Moreover, local changes in the radio topology impacts only locally the virtual structure, improving both the robustness and the stability.

3 Routing Protocols

3.1 Flat Routing

Flat routing protocols do not introduce a hierarchy among the nodes: all nodes are equal and participate to the routing process. Classically, they can be divided in the reactive and the proactive classes.

Reactive. Reactive protocols propose to discover routes *on demand*. In AODV (Ad Hoc On demand Distance Vector Routing) [8], when a node wants to send a `Data Packet`, and no route is present, it sends a `Route Request` in broadcast. When a node receives a `Route Request`, it adds an entry in its routing table pointing to the source of the request via the previous hop. Then, if no route to the searched node is known, it forwards the `Route Request`. Else, it generates a `Route Reply`, sent along the inverse route. The `Route Reply` is forwarded in unicast and creates an entry in the routing table of each intermediary node pointing to the destination. A `Route Error` is created if a route is broken. This packet sent to the source allows to delete failed entries in the routing tables. The route discovering introduces a latency, but the overhead is minimized. Flooding used for the route discovering can create a broadcast storm.

Proactive. Proactive protocols propose to maintain all along the time all the routes. If a node floods periodically topology packets in the network, an heavy load on the radio medium is created and collisions occur. OLSR (Optimized Link State Routing) [5] proposes to limit the impact of the flooding. Each node selects a Multi Point Relays (MPR) set: the MPRs are neighbors and cover all the 2-neighborhood. When a node sends a `Topology Packet`, only its MPRs forward it. Recursively, only the MPR of the MPR will relay the control packets, reducing the overhead. However, the overhead remains important and could be useless if a node communicates only with a few destinations, using only a few routes.

3.2 Hierarchical Routing

To overcome the problems of flat routing, clustered routing protocols were proposed. CBRP (Cluster Based Routing Protocol) [6] is based on a hierarchy of clusters and clusterheads. Each node sends periodically `Hellos` containing the list of neighbors and the list of adjacent clusters (the clusterheads of neighbors). The protocol is reactive: a node sends a route discovering among the clusterheads and the gateways. The reply is sent along the inverse route, each clusterhead trying to bypass itself in the route if possible. The route is registered in the packet, CBRP being a source routing protocol. However, the topology is only used for the route discovering. CBRP is finally a flat routing protocol, the route being constituted only by a list of individual nodes.

3.3 Routing on a Self-organization

[11] proposed a leader-based framework of routing protocol, VSR (Virtual Structure Routing), based on the virtual structure of self-organization described in the section 2.2. A proactive routing protocol is implemented in a cluster: each node knowns proactively all the routes of its cluster. For the inter-cluster routing, a route is discovered on demand and is based on the cluster id rather than on the node id in order to benefit of the virtual structure stability and robustness properties. A route is constituted by a list of clusters from the source to the destination. When a node wants to send a `Data Packet` and no route is known, it sends a `Route Request` to the nearest backbone member. This backbone member adds its cluster id in the route contained in the header of the packet and forwards the packet in multicast to other backbone members. When a backbone member receives a request and the destination is unknown, it adds its cluster id in the route if it is not present and forward the packet. In the other case, if the

destination is present either in the neighborhood table or in the routing table, it generates a `Route Reply`. Then, the `Route Reply` is forwarded in unicast like `Data Packets`: a forwarder tries to reach the cluster of the route, nearest of the destination. A cluster is reachable according to the following criteria:

- A neighbor is in the searched cluster
- A neighbor is gateway for the searched cluster
- A node in the cluster is gateway for the searched cluster. This node is reachable thanks to the intra-cluster routing protocol

In consequence, the route length is optimized thanks to the local knowledge of the cluster and neighbors. Finally, a route repair mechanism is proposed: the routing algorithm is re-executed forbidding the previously failed node.

4 Performances Comparison

4.1 Simulation Guidelines

We present here results about simulations using OPNET Modeler. We used the 802.11b model proposed in OPNET with a standard 300m radio range, in DCF mode, without RTS/CTS. Each node moves itself according to the random waypoint mobility model, without any pause time. All results are computed with a 95% confidence interval. We consider as standard a mobility of 5m.s^{-1}, 40 nodes, a degree (number of neighbors) of 10 , and 4 simultaneous flows. Each simulation lasts 600 seconds. The traffic generation is modeled as follows: flows of 20 `data packets` interspaced by 0.25s are sent. For each flow, a destination and a source are randomly chosen. The inter-flow time follows an exponential distribution centered on 5 seconds to have on average the chosen number of simultaneous flows. The packet size follows an exponential distribution centered on 128 octets.

The results highlight the pertinence of using a self-organization to provide a routing scheme based on a virtual topology. We compare the performances of VSR with the performances under the same conditions of AODV, CBRP and OLSR. We simulated VSR with $k_{cds} = 1/k_{cluster} = 2$ and $k_{cds} = 2/k_{cluster} = 3$. Both configurations present similar results compared to other routing protocols, except for the overhead. Hence, we chose to represent the results only for $k_{cds} = 1/k_{cluster} = 2$, except in the section dealing with the overhead.

4.2 Performances

Horizontal Scalability. We investigate the horizontal scalability of the different routing protocols (fig. 2). The network cardinality comprises 20 to 90 nodes. To study uniquely the impact of the number of nodes, we maintain the degree as constant. OLSR presents the lowest delay since it is a proactive protocol, and a route is found immediately. Oppositely, CBRP and AODV which are reactive present an higher delay (30ms are required when 90 nodes are present). VSR

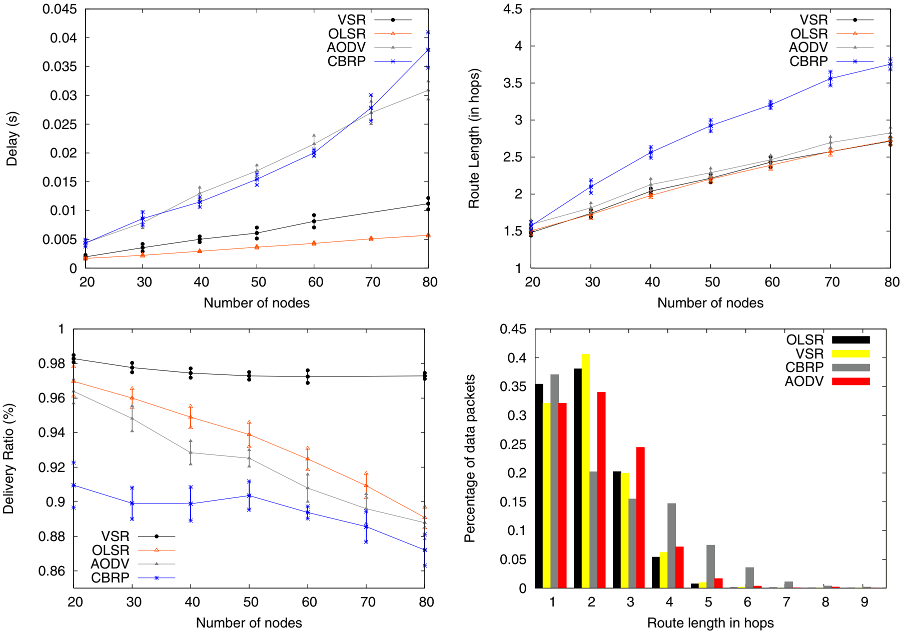


Fig. 2. Horizontal Scalabalility and the route length distribution

benefits from the self-organization: the delay is higher than OLSR, but the difference remains reasonable, even with 80 nodes. The length of the routes discovered by CBRP are longer: the `Route Requests` pass through the clusterheads and gateways topology. Besides, the route shortening when a clusterhead forwards a `Route Reply` seems to construct routes longer than the shortest routes. OLSR constructs shortest routes since each node has a complete knowledge of the topology. AODV, in spite of its reactive behavior, constructs routes near from shortest routes. VSR benefits from the virtual topology and does not create longer route although a hierarchy is used. The distribution of the route length is reported in fig.2: the proportion of the number of routes which are equals to x hops is reported (x varying from 1 to 9 hops). OLSR and VSR present a very similar distribution. Since OLSR computes shortest paths, VSR achieves the discovering of short routes. AODV discovers sometimes longer routes, but the distribution is analogous to OLSR and VSR. CBRP discovers the longest routes, creating potentially more load and collisions. The delivery ratio of CBRP is the lowest (90% with 50 nodes or more): the route is constituted by a list of nodes and is not robust. The hierarchy seems not fully exploited. AODV is a reactive protocol, and sometimes collisions occur for the `Route Requests`. No route is in this case discovered, and some `Data Packets` are dropped. In the same way, `Topology Packets` for OLSR can collide, creating a lack of some routes in the network. These collisions are more frequent when more nodes are present in the network: the delivery ratio decreases when the number of nodes increases. VSR combines the approaches thanks to the self-organization. Moreover, the backbone allows

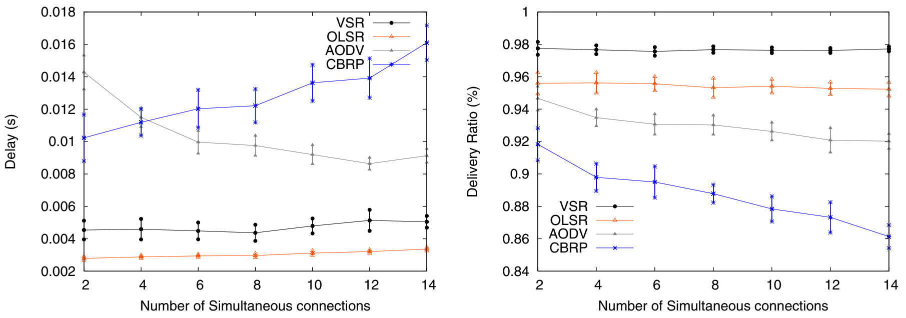


Fig. 3. Vertical Scalability

to save transmissions of control packets: collisions are less frequent. Thus, the delivery ratio is improved, and remains very stable according to the number of nodes.

Vertical Scalability. VSR and OLSR present the lowest delay, invariant according to the number of simultaneous connections (fig. 3). The delay of AODV decreases when the number of connections increases: more connections are initiated, creating more route discoverings. In consequence, a `Data Packet` has an higher probability to have already a route in its routing cache toward the searched destination. The delay of CBRP increases when the load in the network increases: more collisions occur because of its long routes and its unoptimized route discovering. For the route discovering, on average 20ms are required for VSR, 50ms for AODV and 100ms for CBRP. The backbone helps greatly to optimize the route discovering. The delivery ratio of AODV, OLSR and VSR seems scalable according to the load of the network. VSR keeps on presenting the highest delivery ratio according to its stable routes thanks to the virtual topology. OLSR presents a lower delivery ratio, but stable according to the load. The delivery ratio of AODV seems to decrease slightly because of the collisions of the `Route Requests`.

Mobility. The impact of the mobility is represented in figure 4. The delay of OLSR remains the lowest: no delay is required to maintain or to construct a route. VSR presents a delay almost stable. The delay of AODV is higher, although almost stable. CBRP seems to suffer more from topology changes. The delivery ratio of CBRP decreases quickly when the mobility increases: at 25m.s^{-1}, 80% of the packets are received by the destination. VSR, AODV and OLSR are more scalable and present an higher delivery ratio. The packet losses increase, since rapid topology changes create route breaks. For reactive protocols, some `Data packets` are dropped and a new route discovering is initiated. For proactive protocols, a node must wait the next `Topology Packets`. VSR presents the highest delivery ratio: routes are a list of clusters, and the route of individual nodes is updated on the fly, according to the local knowledge. The routes of VSR seem more robust due to the robustness of the virtual topology.

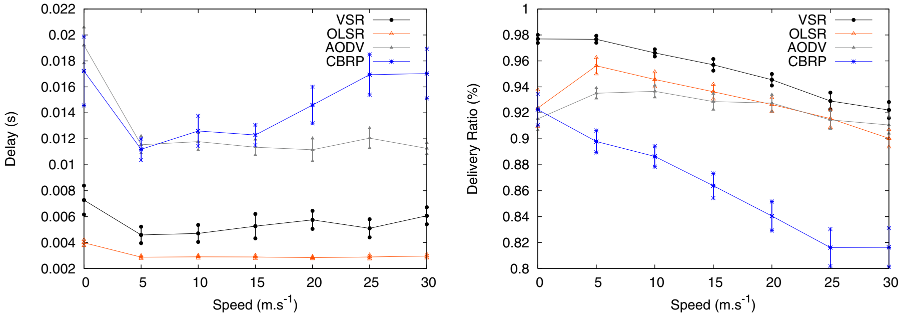


Fig. 4. Impact of the mobility

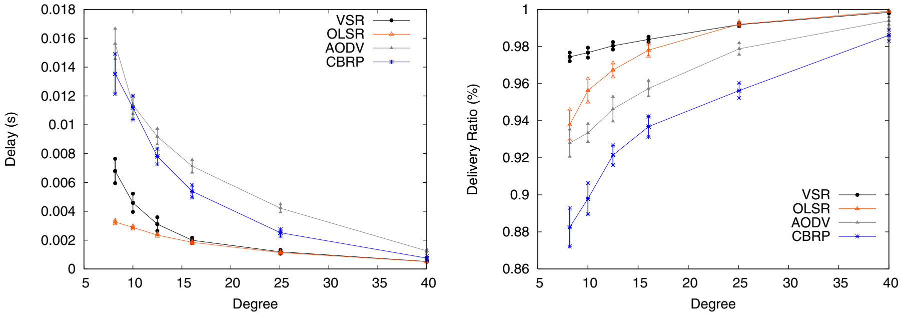


Fig. 5. Impact of the density

Density. Finally, the impact of the degree, i.e. the number of neighbors, is investigated (fig. 5). We note that the delay and the packet loss decreases when the density increases: the radius of the network is smaller, and routes are on average smaller. When the network is a single hop, all the protocols seem to react efficiently. However, for low density, reactive routing protocols seem to lose packets and suffer from an higher delay. This phenomena is perhaps because the broadcast is not reliable and many `Route Requests` are lost since the network is very sparse. The delay of VSR increases for very sparse networks since the `Route Request` must be forwarded farther. Nevertheless, the delivery ratio of VSR remains the highest among all the routing protocols: routes are stable, and the protocol seems to react efficiently to the lack of reliability of broadcasts thanks to the flooding through the virtual structure.

Overhead. Under the previous assumptions, the overhead of VSR is mainly constituted by the hellos (50%) and the maintenance of the virtual structure (40%). The route discovering is optimized thanks to the self-organization structure. The overhead of OLSR is mainly driven by the topology packets (78%): a topology packet must be flooded in the whole network. The overhead of AODV is constituted by `Route Request` (64%) and `Route Reply` (36%). The route discovering, because of the flooding, presents an important cost. Finally, the `Route Request`

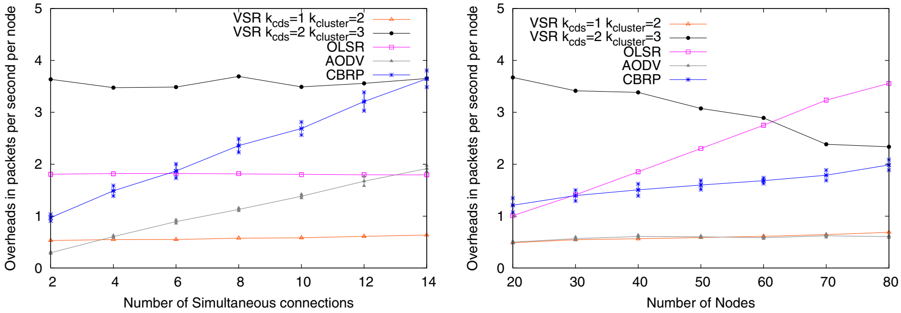


Fig. 6. Overheads of the different routing protocols

represent 90% of the overhead of CBRP. The route discovering seems not efficiently exploit the hierarchy in the network.

We also compute the total overheads of the different routing protocols according to the load of the network in packet per second per node (fig. 6). We can remark that the overhead of OLSR is stable: the proactive overhead is independent of the volume of the traffic. The overhead of AODV increases when more `Data Packets` must be sent. OLSR overpasses AODV when more than 14 simultaneous connections are present in the network. The overhead of CBRP increases quicker than AODV. VSR is an hybrid routing protocol and takes advantage efficiently of the virtual structure to limit the flooding. In consequence, VSR with $k_{cds} = 1/k_{cluster} = 2$ presents the lowest overhead when the load of the network exceeds a threshold. When, $k_{cds} = 2/k_{cluster} = 3$ the overhead is important because of the proactive routing protocol inside each cluster. Thus, OLSR should be implemented to reduce the control traffic. Besides, we can remark that AODV presents an efficient route discovering: if the number of connections is constant, the overhead because of the route discovering does not increase importantly. VSR is also very scalable. The route discovering process of CBRP is less efficient and presents an higher overhead. However, the overhead of OLSR increases quickly: more nodes must send `Topology Packets`, even if the number of connections remains constant.

Route Repairs. Finally, the impact of the route repair mechanism is measured (fig. 7). For the sake of the genericness, we do not assume the existence of a cooperation between the MAC layer and the ad hoc routing protocol layer. In the same way, the promiscuous mode is considered as non available. An `Acknowledgment packet` is in consequence required for each data packet. If no `Acknowledgment` is received after 3 transmissions, a route repair is initiated.

Route repairs introduce timeout mechanisms and retransmissions. Thus, the delay is increased for both CBRP and VSR. However, since CBRP presents already an higher delay than VSR without route repairs, CBRP keeps on presenting an higher delay. Oppositely, the delivery ratio is greatly improved. At 5m.s^{-1}, the delivery ratio is improved by 6% for CBRP. VSR presents the highest delivery ratio, superior to 99% even with a speed of 25m.s^{-1}. This improvement

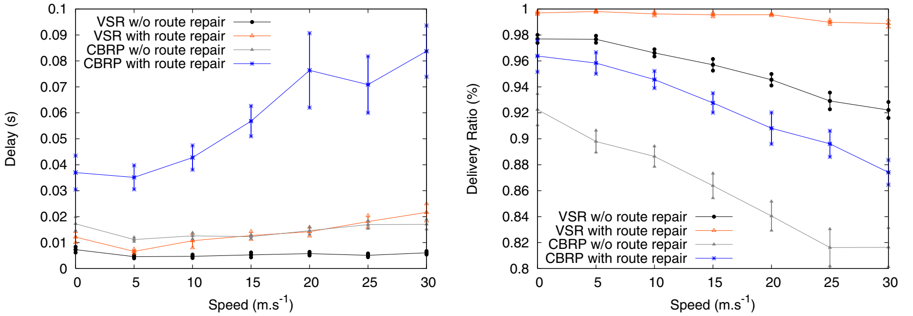


Fig. 7. Impact of the route repair mechanism of CBRP and VSR

introduces naturally an overhead: at least an `Acknowledgement` packet is required for each `Data Packet`. However, if a promiscuous mode is available, the overhead is either negligible. In a cross-layer approach with a MAC notification, it is even null.

5 Conclusion

In this paper, we compare the performances of a routing protocol (VSR) based on a self-organization scheme with the more classical flat approaches. VSR is based on self-organization paradigms and benefits of the stability and robustness properties of the virtual structure. It allows to offer to the routing protocol a hierarchical view of the network. VSR uses a pro-active routing inside the clusters and a reactive one for inter-cluster routing. Simulation results highlight the behavior of VSR. VSR improves the vertical and horizontal scalabilities; moreover, less packets are dropped, and the delay does not seem to suffer from the hierarchical structure. In fact, VSR appears to be a very interesting way to optimize the trade-off of proactive and reactive protocols. Based on this work, it should be interesting to use OLSR for intra-cluster routing to reduce the overhead. We are mainly interested by two perspectives. First, we have developped a testbed to validate our approach and the first results we get are very relevant. Second, because self-organization deals with better nodes and local decisions, the capacity in terms of flow should be quantified.

References

1. K. M. Alzoubi, P.-J. Wan, and O. Frieder. Distributed heuristics for connected dominating set in wireless ad hoc networks. *IEEE/KICS Journal of Communications and Networks*, 4(1):22–29, march 2002.
2. A. Amis, R. Prakash, T. Vuong, and D. Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *INFOCOM*, Tel-Aviv, Israel, March 1999. IEEE.
3. M. Cardei, X. Cheng, X. Cheng, and D.-Z. Du. Connected domination in ad hoc wireless networks. In *International Conference on Computer Science and Informatics (CSI)*, North Carolina, USA, March 2002.

4. B. N. Clark, C. J. Colburn, and D. S. Johnson. Unit disks graphs. *Discrete Mathematics*, 86:165–177, December 1990.
5. T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626, IETF, October 2003.
6. M. Jiang, J. Li, and Y. C. Tay. Cluster based routing protocol (CBRP). Internet draft version 01, IETF, July 1999.
7. C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7):1265–1275, 1997.
8. C. E. Perkins, E. M. Belding Royer, and S. R. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, IETF, July 2003.
9. F. Theoleyre and F. Valois. A virtual structure for mobility management in hybrid networks. In *Wireless Communications and Networking Conference (WCNC)*, Atlanta, USA, March 2004. IEEE.
10. F. Theoleyre and F. Valois. About the self-stabilization of a virtual topology for self-organization in ad hoc networks. In LNCS, editor, *Self-Stabilization Symposium (SSS)*, volume 3764, Barcelona, Spain, October 2005. IEEE.
11. F. Theoleyre and F. Valois. Virtual structure routing in ad hoc networks. In *International Conference in Communications (ICC)*, Seoul, Korea, May 2005. IEEE.
12. J. Wu and H. Li. Dominating-set-based routing in ad hoc wireless networks. In *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL'M)*, Seattle, USA, August 1999. ACM.

Performance Modeling of Epidemic Routing

Xiaolan Zhang[1], Giovanni Neglia[2], Jim Kurose[1], and Don Towsley[1]

[1] University of Massachusetts, Amherst MA 01002, USA
{ellenz, kurose, towsley}@cs.umass.edu
[2] Università degli Studi di Palermo
giovanni.neglia@tti.unipa.it

Abstract. In this paper, we develop a rigorous, unified framework based on Ordinary Differential Equations (ODEs) to study epidemic routing and its variations. These ODEs can be derived as limits of Markovian models under a natural scaling as the number of nodes increases. While an analytical study of Markovian models is quite complex and numerical solution impractical for large networks, the corresponding ODE models yield closed-form expressions for several performance metrics of interest, and a numerical solution complexity that does not increase with the number of nodes. Using this ODE approach, we investigate how resources such as buffer space and power can be traded for faster delivery, illustrating the differences among the various epidemic schemes considered. Finally we consider the effect of buffer management by complementing the forwarding models with Markovian and fluid buffer models.

Keywords: Delay tolerant networks, wireless ad hoc networks, epidemic routing, performance modeling, ordinary differential equations.

1 Introduction

Epidemic routing [13] has been proposed as an approach for routing in sparse and/or highly mobile networks in which there may not be a contemporaneous path from source to destination (i.e., a special case of Delay Tolerant Network). Epidemic routing adopts a so-called "store-carry-forward" paradigm – a node receiving a packet buffers and carries that packet as it moves, passing the packet on to new nodes that it encounters. Analogous to the spread of infectious diseases, each time a packet-carrying node encounters a new node that does not have a copy of that packet, the carrier is said to *infect* this new node by passing on a packet copy; newly infected nodes, in turn, behave similarly. The destination receives the packet when it first meets an infected node. Epidemic routing is able to achieve minimum delivery delay at the expense of increased use of resources such as buffer space, bandwidth, and transmission power. Variations of epidemic routing have recently been proposed that exploit this trade-off between delivery delay and resource consumption, including K-hop schemes [9, 3], probabilistic forwarding [8, 4], and spray-and-wait [12, 11].

Early efforts evaluating the performance of epidemic routing schemes used simulation [13, 5, 8]. More recently, Markovian models have been developed to study the performance of epidemic routing [10, 3, 4], 2-hop forwarding [3], and

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 827–839, 2006.

spray-and-wait [12, 11]. Recognizing the similarities between epidemic routing and the spread of infectious diseases, [10] used ordinary differential equation (ODE) models adapted from infectious disease-spread modeling [2] to study the source-to-destination delivery delay under the basic epidemic routing scheme, and then adopted Markovian models to study other performance metrics.

In this paper, we develop a rigorous, unified framework, based on Ordinary Differential Equations (ODE), to study epidemic routing and its variations. The starting point of our work is [3], where the authors consider common node mobility models (e.g., random waypoint and random direction mobility) and show that nodal inter-meeting times are nearly exponentially distributed when transmission ranges are small compared to the network's area, and node velocity is sufficiently high. This observation suggests that Markovian models of epidemic routing can lead to quite accurate performance predictions; indeed [3] develops Markov chain models for epidemic routing and 2-hop forwarding, deriving the average source-to-destination delivery delay and the number of extant copies of a packet at the time of delivery. An analytical study of such Markov chain models is quite complex for even simple epidemic models, and more complex schemes have defied analysis thus far. Moreover, numerical solution of such models becomes impractical when the number of nodes is large.

We develop ODEs as a fluid limit of Markovian models such as [3], under an appropriate scaling as the number of nodes increases. This approach allows us to then derive closed-form formulas for the performance metrics considered in [3], obtaining matching results. More importantly, we are also able to use the ODE framework to further model the so-called "recovery process" (packet deletion at infected nodes, following the successful delivery to the destination), to study more complex variants of epidemic routing, and to model the performance of epidemic routing with different buffer management schemes under buffer constraints. While different recovery processes are studied also in [10, 11] using Markov chains, model simulation is first needed to determine a number of model parameters. Many of our ODE models can be analytically solved, providing closed-form formulas for the performance metrics of interest; in cases where we resort to numerical solution, the computation complexity does not increase with the number of nodes. The drawback of our ODE models is that they are used to evaluate the moments of the various performance metrics of interest, while numerical solution of Markov chain models can provide complete distributions (e.g., for the number of packet copies in the system). Simulation results show good agreement with the predictions of our ODE models.

Through our modeling studies, we obtain insights into different epidemic routing schemes. In particular, we identify rules of thumb for configuring these schemes, we show the existence of a linear relation between total number of copies sent and the buffer occupancy under certain schemes, and we demonstrate that the relative benefit of different recovery schemes depends strongly on the specific infection process. Finally our analysis of buffer-constrained epidemic routing suggests that sizing node buffers to limit packet loss is not vital as long as appropriate buffer management schemes are used.

The remainder of this paper is structured as follows. Basic epidemic routing and our basic ODE model are described and derived in Section 2, allowing one to characterize the source-to-destination delivery delay, the number of copies made for a packet, and the average buffer occupancy. In Section 3, the model is extended for three important variations of basic epidemic routing: K-hop forwarding, probabilistic forwarding and limited-time forwarding; we use these extended models to characterize the tradeoff between delivery delay and resource (buffer, power) consumption in Section 4. In Section 5, we integrate the ODE models with Markov and fluid queue models to study the effect of finite buffers, and compare different buffer management strategies. Finally in Section 6 we summarize the paper and discuss about future work. Throughout the paper, we compare our work with related efforts, where appropriate. Due to space constraint some of the derivations are in [14].

2 Basic Epidemic Routing

In this section we develop our ODE model for basic epidemic routing [13], after briefly describing epidemic routing and the scenario we are considering. We then use the model to study three different recovery techniques for limiting the number of packet copies in the network, validating these models against simulation.

We consider a set of $N + 1$ nodes with a finite transmission range moving in a closed area and different source-destination pairs. We say that two nodes "meet" when they come within transmission range of each other, at which point they can exchange packets. Let us focus on a single packet. The analogy with disease spreading is useful in describing epidemic routing. The source of the packet can be viewed as the first carrier of a new disease, the first *infected* node, and it copies the packet to (infects) every node it meets. These new infected nodes act in the same way. As a result, the population of *susceptible* nodes (i.e., nodes without a copy of the packet) decreases over time. Once a node carrying a packet meets the destination, it passes the packet on to the destination, deletes the packet from its own buffer, and retains "packet-delivered" information (an anti-packet) which will prevent it from receiving another copy of this packet in the future; such a node has *recovered* from the disease. We will shortly consider more sophisticated recovery schemes.

Consider now many packets spreading at the same time in the network. We assume that when two nodes meet they can exchange an arbitrary number of packets, and each node has enough buffer to store all packets (the latter assumption is relaxed in Section 5), thus allowing different infections to be considered independently. We also assume a mechanism exists so that nodes never exchange a packet if both nodes are already carrying a copy of that packet.

2.1 ODE Models for Basic Epidemic Routing

As noted earlier, [3] showed that the pairwise meeting time between nodes is nearly exponentially distributed, if nodes move in a limited region (of area, A) according to common mobility models (such as the random waypoint or random direction model [1]) and if their transmission range (d) is small compared to A, and their speed is sufficiently high. The authors also derived the following formula for estimating the pairwise meeting rate β:

$$\beta \approx \frac{2wdE[V^*]}{A}, \tag{1}$$

where w is a constant specific to the mobility models, and $E[V^*]$ is the average relative speed between two nodes. Under this approximation, [3] showed that the evolution of the number of infected nodes can be modeled as a Markov chain.

We introduce our modeling approach starting from the Markov model for simple epidemic routing. Given $n_I(t)$, the number of infected nodes at time t, the transition rate from state n_I to state n_I+1 is $r_N(n_I) = \beta n_I(N - n_I)$, where N is the total number of nodes in the network (excluding the destination). If we rewrite the rates as $r_N(n_I) = N\lambda(n_I/N)(1 - n_I/N)$ and assume that $\lambda = N\beta$ is constant, we can apply Theorem 3.1 in [7] to prove that, as N increases, the fraction of infected nodes (n_I/N) converges asymptotically to the solution of the following equation[1]:

$$i'(t) = \lambda i(t)(1 - i(t)), \text{ for } t \geq 0 \tag{2}$$

with initial condition $i(0) = \lim_{N\to\infty} n_I(0)/N$. The average number of infected nodes then converges to $I(t) = Ni(t)$ in the sense of footnote 3. The following equation can be derived for $I(t)$ from Eq.(2):

$$I'(t) = \beta I(N - I), \tag{3}$$

with initial condition $I(0) = Ni(0)$. Such an ODE, which we have shown results as a fluid limit of a Markov model as N increases, has been commonly used in epidemiology studies, and was first applied to broadcast in mobile ad hoc network in [6], epidemic routing in [10], as a reasonable approximation.

We remark that 1) the initial population of infected nodes must scale with N, and 2) the pairwise meeting rate scales as $1/N$. Eq.(1) also provides insight into the physical interpretation of the meeting rate scaling, in particular one can consider that the area A increases with N, keeping node density constant. In the following we will consider Eq.(3) with initial condition $I(0) = 1$, which corresponds to an initial fraction of infected nodes $i(0) = 1/N$. Despite the "small" number of initial infected nodes, we will see via our simulation results that the approximation is a good one. We also note that Eq.(3), as well as other related equations we will derive shortly, can also be obtained in a different manner from Markovian models by neglecting terms related to higher moments [2, 14].

2.2 Delay Under Epidemic Routing

Let T_d be the packet delivery delay, i.e., the time from when a packet is generated at the source to the time when it is first delivered to the destination, and denote

[1] Formally, $\forall\epsilon > 0$, $\lim_{N\to\infty} \text{Prob}\{|\sup_{s\leq t}\{n_I(s)/N - i(s)\}| > \epsilon\} = 0$.

its Cumulative Distribution Function (CDF) by $P(t) = Pr(T_d < t)$. Under the same scaling and approximations considered earlier, we can derive the following equation for $P(t)$: $P'(t) = \lambda i(1 - P)$ [14], and in a similar manner

$$P'(t) = \beta I(1 - P). \tag{4}$$

Eq.(4) was proposed in [10], based on an analogy with a Markov process. Solving Eq.(3) and Eq.(4) with $I(0) = 1, P(0) = 0$, we get

$$I(t) = \frac{N}{1 + (N-1)e^{-\beta N t}}, \qquad P(t) = 1 - \frac{N}{N - 1 + e^{\beta N t}}$$

From $P(t)$, the average delivery delay can be explicitly found in closed form as $E[T_d] = \int_0^\infty (1 - P(t))dt = \ln N/(\beta(N-1))$. The average number of copies of a packet in the system when the packet is delivered to the destination, $E[C_{ep}]$, can also be derived [14]: $E[C_{ep}] = \int_0^\infty I(t)P'(t)dt = \frac{N-1}{2}$.

We note that while [3] obtained the same result for the number of copies, derived the Laplace-Stieltjes Transform (LST) of the delay, and from the LST found the following asymptotic expression for the average delay as $N \to \infty$: $\frac{1}{\beta(N-1)}(\ln N + \gamma + O(\frac{1}{N}))$, the derivation is much simpler using our ODE model.

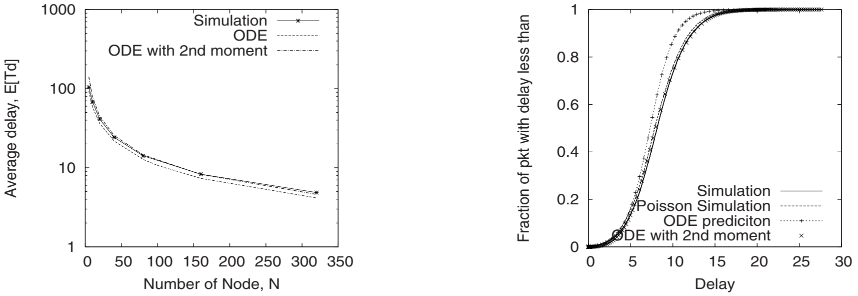
2.3 Recovery from Infection

Next we study the total number of packet copies sent, and the packet's average storage requirement under the recovery schemes proposed in [4]. Clearly, once a node delivers a packet to the destination, it should delete the packet from its buffer to save storage space and prevent the node from infecting other nodes. Moreover, to avoid being reinfected by the packet, a node can store a so-called "anti-packet" once it delivers a packet to the destination. We refer to this scheme as IMMUNE scheme. A more aggressive approach towards deleting obsolete copies is to propagate anti-packets among nodes. An anti-packet can be propagated only to infected nodes (which we will term as IMMUNE_TX scheme), or to both infected and susceptible nodes (VACCINE scheme).

Similar to our earlier analysis in Section 2.1, we can derive ODEs that take into account the recovery process as the limit of Markov models [14], with the additional consideration that we need to scale the number of destinations n_D in a manner similar to the scaling of the number of initially infected nodes, i.e. $\lim_{N\to\infty} n_D/N = d$. For example, if we consider the IMMUNE scheme, the number of infected and recovered nodes should be respectively close to $I(t)$ and $R(t)$, which are solutions of the following equations:

$$I'(t) = \beta I(N - I - R) - \beta I D, \qquad R'(t) = \beta I D$$

where D is the number of destinations and we consider $I(0) = 1, R(0) = 0$, $D = 1$. This model allows us to evaluate the average number of times that a packet is copied during its lifetime, $E[G_{ep}]$. In fact $E[G_{ep}] = \lim_{t\to\infty} I(t) + R(t) - I(0) - R(0)$ and a good approximation can be found through the previous equations by expressing I as a function of R, without the need to solve



(a) Average delay for different N

(b) CDF of delay with $N = 160$

Fig. 1. Delay under epidemic routing

for $I(t)$ and $R(t)$ (see Table 1 for results and [14] for a detailed derivation). Analogous ODEs can be derived for the IMMUNE_TX and VACCINE schemes, and a closed formula can be derived for $E[G_{ep}]$ for the IMMUNE_TX scheme (Table 1). Numerical solutions are needed for the VACCINE scheme.

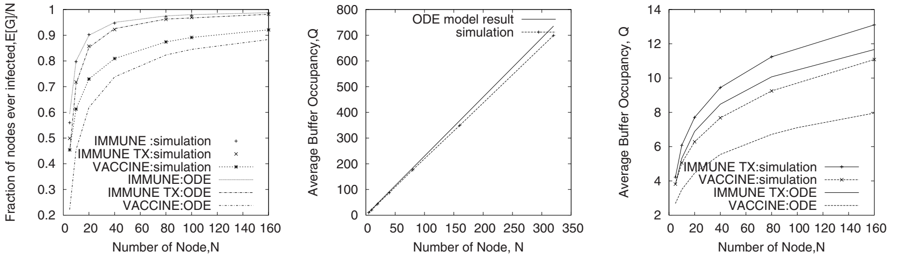
We next consider the average storage requirement in the case of $N+1$ unicast flows, with each node being the source of one flow and destination for one other flow, and each flow generating packets with Poisson rate λ. Denote by L the average packet lifetime (the time from when the packet is generated by the source node to when all copies of the packet are removed from the system). The average number of copies of a packet in the system during its lifetime is given by $\int_0^\infty I(t)dt/L$, where $I(t)$ is the solution to the ODEs that include the recovery process. As the total arrival rate of new packerts to the system is $(N+1)\lambda$, by Little's law, the average number of packets in the system is $(N+1)\lambda L$. Therefore the average total buffer occupancy in the whole network is given by $E[Q_{total}] = (\int_0^\infty I(t)dt/L)(N+1)\lambda L = \int_0^\infty I(t)dt(N+1)\lambda$, and the per-node buffer occupancy is thus $E[Q] = \lambda \int_0^\infty I(t)dt$.

Modeling a node's buffer as an $M/M/\infty$ queue gives the same result [14] and shows a linear relationship between the average buffer occupancy and the number of copies made when IMMUNE is used. In fact, given that each packet is copied $E[G_{ep}]$ times, each flow generates relay traffic of rate $E[G_{ep}]\lambda$, and the total rate of relay traffic in the network is $E[G_{ep}]\lambda(N+1)$ (as there are $N+1$ flows). This traffic is equally divided among the $N+1$ nodes, hence the arrival rate of relay packets to each node is $E[G_{ep}]\lambda$, and the total packet arrival rate is $\lambda(1+E[G_{ep}])$. If a copy is deleted only when the node meets the destination[2], the service rate is $1/\beta$ and the average buffer occupancy is $E[Q] = \frac{\lambda}{\beta}(1+E[G_{ep}])$.

2.4 Model Validation

Throughout the paper, we validate our models using a simulator we developed to simulate the epidemic routing scheme and its variations under various mobility

[2] This is the case under IMMUNE for the basic epidemic routing, and also for two other schemes we are going to consider: probabilistic and K-hop forwarding.



(a) $E[G_{ep}]/N$ (b) Buffer occupancy under IMMUNE (c) Buffer occupancy under IMMUNX TX and VACCINE

Fig. 2. Copies sent and buffer occupancy under epidemic routing

models. Here, we validate our models using a specific setting considered in [3]: nodes move according to random direction model within a 20 × 20 terrain. The transmission range of the node is chosen to be 0.1. The node speed is chosen uniformly in the range 4-10, and the mean trip duration is 1/4. The pair-wise meeting rate for this setting is found to be $\beta = 0.00435$ using the formula in [3].

We vary the number of nodes, N, and let each flow generate packets with Poisson rate $\lambda = 0.01$. The simulation is run long enough so that 100 packets are generated for each flow. The mean and CDF of the delivery delay obtained from the simulation are compared with the model results in Fig.1. We observe that the model is able to accurately predict the delivery delay, capturing the performance trend as N increases, with a slightly larger discrepancy in the CDF. To investigate modeling errors, we run another set of simulations with nodes meeting according to a Poisson process with rate $\beta = 0.00435$ (i.e., we set the meeting rate in the simulation to exactly match the model's meeting rate) and the results of the two sets of simulations are very close (Fig.1.(b)). We thus conjecture that the prediction errors are mainly due to the small number of initial infected nodes. We also use a moment-closure technique to derive a ODE system involving second moments [14]. The modified ODE provides a better prediction of average delivery delay and the CDF of delivery delay (Fig.1).

For the different recovery schemes, Fig.2 plots $E[G_{ep}(N)]/N$, and the average buffer occupancy as predicted by the model and obtained from simulation. We find that the ODE models are more accurate for IMMUNE than for VACCINE. In some sense, any error in the infection process modeling is amplified by the exponentially fast recovery of VACCINE. We observe that IMMUNE_TX only slightly reduces the number of copies sent for each packet, while VACCINE further reduces the number of copies sent. The reduction in buffer requirements is similar for IMMUNE_TX and VACCINE.

3 Extended Model

Although the recovery schemes discussed in the previous section lead to substantial differences in buffer and power requirements, they all achieve the minimum

Table 1. Summary of closed-form expressions obtained for different schemes

Schemes	$I(t)$ $P(t)$	$E[T_d]$	C,G
Epidemic	$I(t) = \frac{N}{1+(N-1)e^{-\beta N t}}$ $P(t) = 1 - \frac{N}{N-1+e^{\beta N t}}$	$\frac{\ln N}{\beta(N-1)}$	$C = \frac{N-1}{2}$, $G \approx N-1$ (IM) $G = \frac{N-3+\sqrt{N^2-2N+5}}{2}$ (IM_TX)
2-hop	$I(t) = N - (N-1)e^{-\beta t}$ $P(t) = 1 - e^{N-1-\beta N t-(N-1)e^{-\beta t}}$	$\frac{1}{\beta}\sqrt{\frac{\pi}{2}}\frac{1}{\sqrt{N-1}}$	$C = \sqrt{\frac{\pi}{2}}\sqrt{N}$, $G = \frac{N-1}{2}$
Prob. Fwding	$I(t) = \frac{N}{1+(N-1)e^{-p\beta N t}}$ $P(t) = 1 - (\frac{N}{N-1+e^{p\beta N t}})^{1/p}$	$[\frac{\ln(N)}{\beta(N-1)}, \frac{\ln(N)}{\beta p(N-1)}]$	$C = \frac{p(N-1)}{1+p}$

delay. The following schemes allow one to trade-off timely delivery with resource consumption.

K-Hop forwarding: Under K-hop forwarding, a packet can traverse at most K hops to reach the destination. We can model these schemes by introducing $K-1$ ODEs, describing the evolution of the number of nodes infected by i-hop paths for $1 \leq i < K$. For example in 2-hop forwarding only the source node can copy the packet to nodes other than the destination, hence the packet spreading rate equals to the rate that the source node meets susceptible nodes. This leads to the following equation for $I(t)$: $I'(t) = \beta(N-I)$ with $I(0) = 1$.

Probabilistic forwarding: Under probabilistic forwarding, each relay node accepts a packet forwarded from an infected node with probability p, resulting in an effective infection rate of $p\beta$, so we have $I'(t) = \beta p I(N-I)$ with $I(0) = 1$.

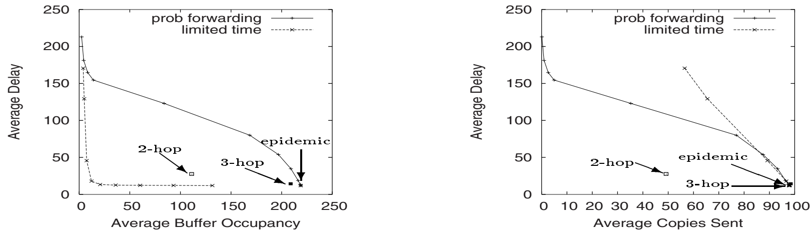
Limited-Time forwarding: For limited-time forwarding, when a *relay* node accepts a packet copy from an infected node, it starts a timer with an exponential random timeout value with mean $1/\mu$. When the timer expires, the node deletes the copy and stores the corresponding anti-packet so that it will not be infected by that packet again. Let $I_r(t)$ be the average number of infected *relay* nodes at time t, and $T(t)$ be the number of timed-out nodes at time t, then we have:

$$I_r'(t) = \beta(I_r+1)(N-I_r-T-1) - \mu I_r, \qquad T'(t) = \mu I_r$$

with $I_r(0) = 0, T(0) = 0$. A variant of this scheme is studied in [14].
With the above ODEs, the packet delivery delay is then found by $P'(t) = \beta I(1-P)$ for 2-hop and probabilistic forwarding, and $P'(t) = \beta(I_r+1)(1-P)$ for limited-time forwarding, both with $P(0) = 0$. We solve the above ODEs either analytically or numerically, and then extend them to consider the recovery process. We also perform simulations to validate the models [14]. The main results are summarized in Table 1.

4 Performance Trade-Off

In this section, we use our ODE models to quantitatively explore the performance trade-offs offered by the various epidemic routing schemes. Previous work [4, 11]



(a) delay vs buffer occupancy tradeoff (b) delay vs number of copies sent tradeoff

Fig. 3. Comparison with IMMUNE recovery

investigated the buffer-delay trade-off by varying the number of nodes. However, we believe that the number of nodes is often given, and it is consequently more important to evaluate the performance trade-offs achieved by different schemes and/or understand how performance changes when configurable parameter values change. In terms of the power-delay trade-off, previous work [11, 12] only considered the trade-off achieved by a special scheme that enforces a fixed number of copies (and hence energy consumption). Our results are mainly based on numerical solution of ODE (for $N = 100$, $\beta = 0.00435$, $\lambda = 0.01$), but also on the asymptotic formulas we derived.

Fig.3.(a) and Fig.3.(b) respectively plot the delay-versus-buffer-occupancy and the delay-versus-number-of-copies-sent trade-offs for the IMMUNE scheme. In the figure, the two singleton points correspond to 2-hop and 3-hop forwarding, while two curves have been obtained for probabilistic forwarding and limited-time forwarding (without reinfection) respectively; for these curves, each point corresponds to a different value of the copy probability, p, and the mean timeout interval, $1/\mu$ (the values are shown in Table 2).

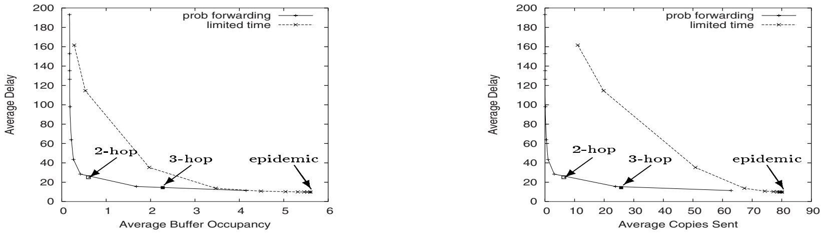
Table 2. Settings considered for Limited-Time and Probabilistic forwarding

Timeout ($1/\mu$)	1	2	5	10	20	40	80	160	320	
Probability (p,%)	0.1	0.5	0.8	1	2	5	10	20	50	80

Let us first consider the delay-versus-buffer-occupancy trade-off. One can reduce the buffer occupancy by decreasing p or $1/\mu$, but at the same time the delay will increase[3]. Limited-time forwarding appears to be the best choice when limiting buffer occupancy is the main concern. As a thumb rule, one can choose $1/\mu \approx 2E[T_d]$ ($= 20$ in this specific setting). This choice significantly reduces the buffer occupancy in comparison to basic epidemic routing (about one tenth), with a negligible increase in the delivery time.

Fig. 3.(b) shows that the curves for probabilistic and K-hop forwarding are similar to the delay-versus-buffer trade-off curves. This is due to the proportionality

[3] Intuitively these schemes behave as the original epidemic model as $p \to 1$ and $1/\mu \to \infty$, whereas $p \to 0$ and $1/\mu \to 0$ correspond to a scenario without any relay: the packet is delivered directly from the source to the destination.



(a) delay vs buffer occupancy tradeoff (b) delay vs number of copies sent tradeoff

Fig. 4. Comparison with VACCINE recovery

between number of copies and buffer occupancy we have shown in Section 2.3. We observe that if power saving is of primary concern, then probabilistic forwarding appears to be the best choice. For example, with $p = 0.008$ the average delay is about 30% less in comparison to the non-relaying case, while the average number of copies is only 3.5. K-hop forwarding offers intermediate performance, without sacrificing either of the two metrics.

Fig. 4 shows the same trade-offs for the VACCINE scheme. We observe that for different schemes, different performance improvements are achieved by VACCINE: in particular, the largest improvement is achieved for probabilistic forwarding, followed by K-hop forwarding, and then limited-time forwarding. The relatively small improvement for limited-time forwarding is due to its intrinsic recovery feature: nodes automatically recover as the timer expires and they cannot be reinfected. The explanation is more complex for the probabilistic and K-hop forwarding schemes. Because of the two counteracting processes – the counter-infection due to anti-packets spreading and the ongoing packet infection – the net recovery speed depends not only on the recovery scheme but also on the specific infection process. Given the same average delivery delay, when the recovery process starts, the average number of nodes infected and the current infection rate are higher under probabilistic forwarding (its infection rate is exponential, hence in the long term it is faster than K-hop). For this reason, we expect the IMMUNE recovery process to be significantly “longer” for probabilistic routing, leading to larger buffer occupancy and more copies. Conversely under VACCINE, the recovery process is much shorter, the buffer occupancy is mainly determined by the initial infection process (before the delivery), and the difference becomes much smaller, as shown in Fig.4.

5 Epidemic Routing Under Constrained Buffer

Thus far, we have assumed that each node has sufficient space to store all packets. Realistically, however, mobile nodes often have limit storage due to cost and form factor. Sizing the buffer to limit end-to-end packet losses due to buffer overflow in store-carry-forward networks is hard. For example, [4] studied buffer occupancy variability for the purpose of buffer sizing, but their model required an empirical distribution obtained from simulation. In this section, we examine the performance of epidemic routing when each node can store at most B packets. We consider three buffer management strategies: *(i) droptail* where newly arriving packets are dropped if the buffer is full (previously studied in [13] through simulation), *(ii) drophead* where the oldest packet in the buffer is dropped to accept newly arriving packets, and *(iii)* source-prioritized drophead, *drophead_sp*, which gives priority to packets arriving directly from the source. We describe the model for drophead_sp here; a full analysis can be found in [14].

Under drophead_sp, when a packet arrives to a full buffer, the node discards the oldest relay packet (i.e., a packet it has received from other node) to make space for the new packet. If all buffered packets are source packets, and the arriving packet is a source packet, the oldest source packet is deleted. Relay packets arriving to a buffer filled with source packets are not accepted. Therefore, given P_f, the probability that a node's buffer is filled with source packets, the effective infection rate is then $\beta(1-P_f)$. P_f can be derived by modeling the number of node-buffered source packets as a Markov chain.

As before, we focus on the spreading of a single packet. Let $\overline{G}_{dhs}$ be the average number of copies made for each packet. In the source node, the copy of this packet becomes older at rate λ, the rate at which new source packets arrive. In infected relay nodes, the packet becomes older whenever another packet arrives, with rate $(\overline{G}_{dhs}+1)\lambda$ (this is the total packet arrival rate to a node by an argument similar to that in Section 2.3). Let $I_j^s(t)$ be the probability that the packet is the j-th newest source packet in the source node's buffer, $I_j(t)$ be the average number of infected *relay* nodes where the copy is the j-th newest packet in the buffer, $S(t)$ be the average number of susceptible nodes, and $D(t)$ be the average number of nodes that have dropped the packet. We can then use the following ODEs to model packet spreading:

$$\begin{aligned}
&S'(t) = -\beta(1-P_f)S\sum_i(I_i^s+I_i)\\
&I_1'(t) = \beta(1-P_f)S\sum_i(I_i^s+I_i)-(\overline{G}_{dhs}+1)\lambda I_1\\
&I_j'(t) = (\overline{G}_{dhs}+1)\lambda(I_{j-1}-I_j), \quad 2\le j\le B\\
&{I_1^s}'(t) = -\lambda I_1^s, \qquad\qquad {I_j^s}'(t)=\lambda(I_{j-1}^s-I_j^s), \quad 2\le j\le B\\
&D'(t) = (\overline{G}_{dhs}+1)\lambda I_B+\lambda I_B^s, \qquad P'(t)=\beta\sum_i(I_i^s+I_i)(1-P)
\end{aligned}$$

The sums above are for $1\le i\le B$. The initial conditions are given by: $S(0)=N-1$, $I_1^s(0)=1$, $I_j^s(0)=0$, for $j=2,...B$, $I_k(0)=0$, for $k=1,...,B$, $D(0)=0$, $P(0)=0$. We find $\overline{G}_{dhs}$ by solving the following fixed-point problem using a binary search algorithm: given $\overline{G}_{dhs}$, we numerically solve the corresponding extended ODE model (including the recovery process) and calculate the accumulated amount of flow from state S to I_1, i.e., $\overline{G}_{dhs}$.

We have simulated these schemes, using the same setting as before ($N=100, \lambda=0.01$, $\beta=0.00435$), with different buffer size $B=5,10,20$, and compared our ODE results with simulation. Table 3 tabulates the loss probabilities. We observe that the models provide reasonable loss probability predictions, and accurately reflect the relative performance of the three dropping schemes. The

Table 3. Loss Probability Under Constrained Buffer

Buffer size	simulation/model	droptail	drophead	drophead_sp
5	simulation	0.9696	0.2234	0.0536
	model	0.8544	0.0928	0.0079
10	simulation	0.9471	0.0315	0.0
	model	0.7891	0.0088	0.0
20	simulation	0.899	0.0016	0.0
	model	0.7011	0.0	0.0

shape of the delay distribution probability function is also well-captured by the model [14]. We observe that naive droptail performs poorly. Drophead provides fast infection, as relay packets are always accepted; however, significant packet losses are incurred for $B \leq 10$. With drophead_sp, although the infection spreads slower, more packets are delivered. If the packet rate is so high that the buffer can only hold its own source packets, drophead_sp degenerates to direct source-destination transmission. Note that with infinite buffers, the average buffer occupancy for this setting is over 200 (Fig. 2.(b)). Our results here suggest that similar performance can be achieved by drophead and drophead_sp with a much smaller buffer size, equal to only 20 packets.

6 Summary and Future Work

In this paper, we proposed a unified framework based on ODEs to study the performance of epidemic routing and its variations. Using these models, we obtained a rich set of quantitative results on the delivery delay, number of copies sent, and buffer requirements (and the tradeoffs of these performance metrics) under various schemes. We further considered buffer-constrained case, and showed that with appropriate buffer management schemes, a much smaller buffer can be used with negligible effect on delivery performance. In the future, we plan to investigate schemes for deleting anti-packets and the overhead of anti-packets.

Acknowledgment

This research was supported in part by the National Science Foundation under the Engineering Research Centers Program, Award number EEC-0213747001, and Italian MIUR project Famous. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

1. C. Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. In *ACM SIGMOBILE Mobile Computing and Communications Review*, volume 5, Issue 3, July, 2001.
2. D.J. Daley and J. Gani. *Epidemic Modelling.* Cambridge University Press, 1999.

3. R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. In *Performance*, October 2005.
4. Z. J. Haas and T. Small. A new networking model for biological applications of ad hoc sensor networks. To appear in IEEE/ACM Transactions on Networking.
5. P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS-X*, 2002.
6. A. Khelil, C. Becker, J. Tian, and K. Rothermel. An epidemic model for information diffusion in manets. In *Proceedings of MSWiM*, 2002.
7. T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump markov processes. *Journal of Applied Probabilities*, pages 49–58, 1970.
8. A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *ACM Mobicom (poster session)*, 2003.
9. G. Sharma and R. R. Mazumdar. Delay and capacity tradeoffs for wireless ad hoc networks with random mobility. Submitted for publication.
10. T. Small and Z. J. Haas. The shared wireless infostation model - a new ad hoc networking paradigm. In *Mobihoc*, 2003.
11. T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *ACM workshop on Delay Tolerant Networking*, 2005.
12. T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *ACM workshop on Delay-tolerant networking*, 2005.
13. A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.
14. X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. Technical Report 2005-44, UMASS Computer Science. `ftp://gaia.cs.umass.edu/pub/Zhang05_epidemic_TR.pdf`.

Maximum Lifetime Routing and Data Aggregation for Wireless Sensor Networks

Cunqing Hua and Tak-Shing Peter Yum

Department of Information Engineering,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
{chua0, yum}@ie.cuhk.edu.hk

Abstract. In this paper we solve the maximum lifetime routing (MLR) problem for a sensor network by joint optimizing routing and data aggregation. We present a smoothing method to overcome the nondifferentiability of the objective function. By exploiting the special structure of the network, we derive the necessary and sufficient conditions to achieve the optimality. Based on these conditions, a gradient descent algorithm is designed for its solution. The proposed algorithm is shown to converge to the optimal value efficiently under all network configurations. The incorporation of optimal routing and data aggregation is shown via many examples to provide significant improvement of the network lifetime.

1 Introduction

Energy-efficient routing [1, 2, 3] has long been studied in the context of wireless ad-hoc networks and sensor networks. Its basic idea is to route the packets through the minimum energy pathes so as to reduce the end-to-end energy consumption. But this tends to overload the minimum energy path, causing the nodes on this path quickly run out of battery energy and disconnecting a vital link. This is undesirable in particular for sensor network where sensor nodes are collaborating for common work.

To cope with this problem, many researchers have proposed to study the maximum lifetime problem based on the linear programming formulation [4, 5, 6, 7]. Here, instead of trying to minimize the path energy consumption, the objective is to select the route and the corresponding power levels to maximize the network lifetime. According to the criticality of a specific mission, the network lifetime may have different definitions, such as given by [8]. The key problem that these schemes try to address is how to find the route and the corresponding flow rates without centralized computations.

The above schemes are applicable for both wireless ad-hoc networks and sensor networks. But for sensor networks, an important feature was not considered in these schemes. It is well-known that data collected in the sensor network is spatially correlated, that is, there exists redundancy in the data collected by the neighboring nodes. It is therefore possible to reduce transmission overhead by aggregating the data at the intermediate nodes. Some research efforts have been made to exploit the data correlation feature to improve the performance of

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 840–855, 2006.

various communication protocols [9,10,11,12,13,14]. These work illustrated that data aggregation can greatly improve the performance of various communication protocols(channel coding, routing, MAC, etc.).

In this paper, we present a model to incorporate the routing and data aggregation into the maximum lifetime problem. By taking the data correlation into the consideration, the network lifetime can be extended from two dimensions. One is to aggregate the data at the intermediate nodes so as to reduce the transmission overhead of the nodes near the sink node, the other is to do maximum lifetime routing as done by [5, 6, 7]. However, these two should be considered simultaneously. In our model, we adopt the geometric routing [15] whereby the routing is determined solely according to the node positions, and each node is associated with a set of routing variables denoted as the fraction of traffic towards its downstream neighbors. We formulate the maximum lifetime routing(MLR) problem as an optimal routing problem where the objective is to find the optimal set of variables for each node to maximize the network lifetime. The contribution of this paper includes: (i) Our model allows different data correlation models such as that proposed in [11] to be incorporated without intervening the underlying routing scheme. (ii) We propose a smoothing method to overcome the non-differentiability of the MLR problem, with which we can design a localized algorithm for each node to compute its routing variables without significant message exchanges. The first feature is desirable because data correlation model depends highly on the specific application, so our model allows different correlation models to work with the underlying routing seamlessly, while the second feature is a must for practical implementation of the algorithm in a sensor network with large number of nodes.

In the following, we first present the system models and define the maximum lifetime routing problem in Section 2. In Section 3, we propose a smoothing function for the maximum lifetime routing problem and provide the analytic results on the optimality conditions. Simulation results are presented in Section 4 and finally we conclude this work in Section 5.

2 System Model

We can model the topology of a wireless sensor network as a undirected graph $G(N, A)$, where N is the set of nodes, A is the set of undirected links. A special node $t \in N$ is called the sink node who is the destination of all other nodes. To capture the characteristic of the network, we need to specify, in addition, the routing model, the data correlation and aggregation model and the power consumption model.

2.1 Geometric Routing Model

The routing algorithm suitable for use belongs to the class of *geometric routing* algorithms [16]. Every sensor node is assumed to know its own position as well as that of its neighbors. Each node can forward packets to its neighbor nodes within its transmission range that are closer to the sink node than itself. In essence, using *geometric routing*, node makes routing decisions with only the position information of the involved nodes. It is therefore a localized algorithm and particularly suitable for large sensor networks.

Let N_i denote the set of neighbors of a node i and $N_i = \{j|\, d_{ij} \leq R, j \in N\}$, where d_{ij} is the Euclidean distance of node i and node j, and R is the radius of the transmission range. According to the *geometric routing* rule, only those neighbors that are closer to the sink node t can serve as the downstream nodes. Let us denote this set of downstream neighbors as $S_i = \{k|\, d_{kt} < d_{it},\, k \in N_i\}$. Symmetrically, the set of upstream neighbors is denoted as A_i. Each link between node i and its downstream neighbor $k \in S_i$ has a routing variable ϕ_{ik} to denote the fraction of the aggregated traffic of node i that will be routed through node k. Clearly, the *flow conservation law* requires $\sum_{k \in S_i} \phi_{ik} = 1$.

2.2 Data Correlation and Aggregation Model

The aggregated traffic λ_i of a node i is a superposition of two parts: local traffic generated by the node itself when sensing the surrounding environments, and the transit traffic from upstream nodes. In other words,

$$\lambda_i = r_i + \sum_{j \in A_i} \lambda_j \phi_{ji}, \qquad i = 1, 2, \cdots, N. \tag{1}$$

In sensor networks, data collected at neighboring nodes are often spatially correlated. It is benefit to remove the redundant information collected at upstream nodes to reduce traffic overhead at the downstream nodes. To capture this feature, we adopt a data correlation model similar to that studied in [11]. Specially, if no side information is available from other nodes, the raw data X_j at a node j is originally entropy coded with $H(X_j) = Y$ bits. However, it can be reduced to $H(X_j|X_{i1}, \cdots, X_{ik}) = y \leq Y$ bits at a downstream node i, where $\{X_{i1}, \cdots, X_{ik}\}$ is the set of side information available at node i. We define the correlation coefficient q_{ji} for a node j and node i as $q_{ji} = 1 - H(X_j|X_{i1}, \cdots, X_{ik})/H(X_j)$, and obviously $0 \leq q_{ji} \leq 1$. Then the aggregated traffic after considering the correlation is

$$\lambda_i = r_i + \sum_{j \in A_i} \lambda_j \phi_{ji}(1 - q_{ji}), \qquad i = 1, 2, \cdots, N. \tag{2}$$

2.3 Power Consumption Model

A sensor node consumes power when it is sensing and generating data, receiving, transmitting, or even simply standby. The power e_g for generating one bit of data is assumed to be the same for all nodes. The standby power consumed by a node, again assumed to be the same for all nodes and independent of traffic, is denoted by e_s. For power used in receiving and transmitting, we adopt the *first order radio model* described in [1]. Specially, a node needs $\epsilon_{elec} = 50nJ$ to run the circuitry and $\epsilon_{amp} = 100pJ/bit/m^2$ for the transmitting amplifier. So the power consumed by a node in receiving a unit of data is given by

$$e_r = \epsilon_{elec} \tag{3}$$

and the power consumed in transmitting a unit of data packet to neighbor node j is given by

$$e_{ij} = \epsilon_{elec} + \epsilon_{amp} \cdot d_{ij}^n \tag{4}$$

Here we consider the path loss of exponent n, which is usually $2 \leq n \leq 4$ for the free-space and short-to-medium-range radio communication. The mean power consumption of node i, denoted as w_i, is therefore

$$w_i = e_s + e_g r_i + e_r \sum_{j \in A_i} \lambda_j \phi_{ji} + \lambda_i \sum_{k \in S_i} e_{ik} \phi_{ik} \tag{5}$$

Where the first term is the standby power consumption, the second term is the power for sensing, the third term is the power for receiving and the fourth term is the power for transmitting.

2.4 Maximum Lifetime Problem

Assume each node i has an initial battery energy E_i, the lifetime T_i of node i is defined as the expected time for the battery energy E_i to be exhausted, that is, $T_i = E_i/w_i$ where w_i is given by (5). Similar to [5,6], we define the network lifetime T_{net} as the time that the first node that runs out of energy, that is

$$T_{net} = \min_{i \in N} T_i \tag{6}$$

The power consumption w_i is a function of $\boldsymbol{r}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$. However, the set of aggregated traffic $\boldsymbol{\lambda}$ can be obtained from $\boldsymbol{r}$ and $\boldsymbol{\phi}$ from (2). Therefore, T_{net} depends only on $\boldsymbol{r}, \boldsymbol{\phi}$ and the initial battery energy $\boldsymbol{E}$. If $\boldsymbol{r}$ and $\boldsymbol{E}$ are given, the network lifetime is solely determined by the set of routing variables $\boldsymbol{\phi}$. We therefore state the maximum lifetime routing(MLR) problem as follows:

MLR: *Given the traffic generating rate $\boldsymbol{r} = \{r_i\}$,the initial battery energy $\boldsymbol{E} = \{E_i\}$ and the data correlation coefficient $\boldsymbol{q} = \{q_{ij}\}$, finding a set of routing variable $\boldsymbol{\phi} = \{\phi_{ij}\}$ for a sensor network G(N, A') such that the network lifetime T_{net} is maximized.*

Let $\tilde{w}_i$ denote the normalized power consumption, that is, $\tilde{w}_i = w_i/E_i$. It is obvious that maximizing the network lifetime T_{net} is equivalent to minimize the maximum normalized power consumption $\tilde{w}_i$ for all $i \in N$. We therefore rewrite the MLR problem formally as

$$\begin{aligned} &\text{minimize } \max_{i \in N} \tilde{w}_i \qquad (7)\\ &\text{subject to } \phi_{ij} \geq 0, \sum_{j \in S_i} \phi_{ij} = 1, \forall i. \end{aligned}$$

3 Distributed Solution for the MLR Problem

The max function (7) in the MLR problem is nondifferentiable, so some simple solutions based on the gradient descent methods are not directly applicable. There are many different approaches that have been studied to overcome this difficulty. One is to transform the $\min - \max$ problem to an equivalent optimization problem (e.g. [7]), such that subgradient algorithms can be used to solve the transformed problem. There is also a family of regularization approaches to obtain the smooth approximation for the max function in literature, for example, the entropy type approximation [17,18], the two dimensional approximation [19], etc.. All these approaches are known as a special case of the so-called smoothing method, an overview of these approaches can be found in [20]. In this section, we propose a smoothing function to approximate the max function in MLR problem (7) by exploiting the special structure of the network. We derive the necessary and sufficient conditions that are required to achieve the optimality of the smoothed problem.

3.1 Problem Transformation

Note after applying the geometric routing, the original undirected network $G(N, A)$ is reduced to a *directed acyclic graph(DAG)* $G(N, A')$, where A' is the set of directed links, and sink node t is the root of the DAG. For any such DAG, we can find a *separation* $s = (N_A|N_B|N_C)$ to partition the node set N into three subsets N_A, N_B and N_C, where N_B is the *cut set*, N_A and N_B are two disjoint node sets. Without loss of generality, let the sink node be located in subset N_C.

Normally we can find many such separations given a directed acyclic graph. For a specific *separation*, we can find a set of routing variables $\boldsymbol{\phi}(s)$ for the nodes in N_A and N_B to minimize the maximum energy consumption rate of the subset N_B, which we denote as $\tilde{w}(s) = min\ max\{\tilde{w}_l, l \in N_B\}$. Since there exists multiple separations, we can always find the worst separation s^* which has the largest minimax energy consumption rate $\tilde{w}(s)$ among all possible separations, i.e., $s^* = \arg\max\{\tilde{w}(s_1), \tilde{w}(s_2), \cdots\}$. We call the corresponding cutset N_B^* as the bottleneck set since this is the node set that limits the lifetime of the network. The problem is therefore to find a set of routing variables for the bottleneck nodes to achieve the minimax energy consumption $\tilde{w}(s^*)$. Formally, we have

$$\begin{aligned} &\text{minimize} \quad \max_{l \in N_B^*} \tilde{w}_l \\ &\text{subject to} \;\; \phi_{ij} \geq 0, \sum_{j \in S_i} \phi_{ij} = 1, \forall i. \end{aligned} \tag{8}$$

Note that (8) is similar to (7) except on the following two points:

- First, the size of the problem for (8) is reduced from $|N|$ to $|N_B^*|$, where $|N|$ and $|N_B^*|$ are the size of the set N and N_B^* respectively.
- Second, the values $\tilde{w}_l, l \in N_B^*$ tend to have a small difference between them because they belong to the same cutset.

The max function in (8) is still not differentiable, however, we can approximate it using the smoothing methods taking advantage of the above two points. We define $\mu = \sum_{l \in N_B} \tilde{w}_l / |N_B^*|$ as the mean power consumption of the bottleneck set, and introduce the following smoothing function

$$U = \mu^2 + \frac{c}{|N_B^*|} \sum_{l \in N_B^*} (\tilde{w}_l - \mu)^2 \tag{9}$$

where c is a positive nondecreasing sequence. The smoothing function is a penalty function consisting of two terms. The physical interpretation of the first term of U is to minimize the mean power consumption of bottleneck nodes. This is achieved by aggregating the data at upstream as much as possible so as to reduce the overall traffic across the bottleneck nodes. The second term of U can be understood as a penalty to the total variability of $\tilde{w}_l$s, which is achieved by optimal routing to equalize the power consumption of the set of bottleneck nodes.

3.2 Optimality Conditions

In this section, we discuss the necessary and sufficient conditions that must be satisfied to achieve the optimality of the smoothed problem (9). Using the routing variables as the control variables, we extend the techniques in [21] to derive the necessary and sufficient conditions for the optimality. Note that the discussion of this section is applicable to non-bottleneck node cuts as well. So without abusing the notations, we use N_A, N_B and N_C to denote three corresponding subsets.

First of all, we can rewrite the smoothing function (9) as

$$\begin{aligned} U &= \mu^2 + \frac{c}{|N_B|} \sum_{l \in N_B} (\tilde{w}_l - \mu)^2 \\ &= \mu^2 + \frac{c}{|N_B|} \left(\sum_{l \in N_B} \tilde{w}_l^2 - 2\mu \sum_{l \in N_B} \tilde{w}_l + |N_B| \mu^2 \right) \\ &= \frac{c}{|N_B|} \sum_{l \in N_B} \tilde{w}_l^2 - \frac{(c-1)}{|N_B|^2} \left(\sum_{l \in N_B} \tilde{w}_l \right)^2 \end{aligned}$$

Here we use the fact that $\mu = \sum_{l \in N_B} \tilde{w}_l / |N_B|$. Since all $\tilde{w}_l$s are the function of the routing variable $\boldsymbol{\phi} = \{\phi_{ij}\}$, we can differentiate U directly from the above equation as

$$\begin{aligned} \frac{\partial U}{\partial \phi_{ik}} &= \frac{2c}{|N_B|} \sum_{l \in N_B} \tilde{w}_l \frac{\partial \tilde{w}_l}{\partial \phi_{ik}} - \frac{2(c-1)}{|N_B|^2} \sum_{l \in N_B} \tilde{w}_l \sum_{l \in N_B} \frac{\partial \tilde{w}_l}{\partial \phi_{ik}} \\ &= \frac{2}{|N_B|} \sum_{l \in N_B} \left(c\tilde{w}_l - (c-1)\mu \right) \frac{\partial \tilde{w}_l}{\partial \phi_{ik}} \end{aligned} \tag{10}$$

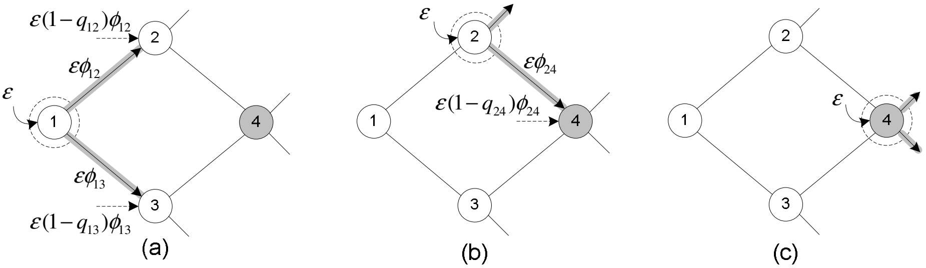


Fig. 1. Three possible relations of node i, k and l: (a) non-adjacent source node and bottleneck node; (b) adjacent source node and bottleneck node; (c) co-located source node and bottleneck node

So what we need to do is to find $\partial\tilde{w}_l/\partial\phi_{ik}$. To this end, we introduce a set of dummy variables $r_i(i \in N)$, where r_i can be interpreted as the dummy traffic injected into node i. This dummy traffic r_i follows the same set of routing given by $\boldsymbol{\phi}$. To derive $\partial\tilde{w}_l/\partial\phi_{ik}$, we need to consider three possible relations of the source node i, its next-hop neighbor k and the bottleneck node l. A simple four-node topology is shown in Fig. 1 to illustrate these three possible scenarios.

(a) **Non-adjacent source node**

If the source node i is not adjacent to the bottleneck node l, let us consider a small increment ϵ to the input rate r_i, this will cause an increment $\epsilon\phi_{ik}$ to the traffic rate of its next-hop neighbor k. Taking into account the data correlation between node i and k, the amount of increment is reduced to $\epsilon(1 - q_{ik})\phi_{ik}$ from node k downwards. Since node k is not a bottleneck node, this extra traffic is equivalent to an increment of $\epsilon(1 - q_{ik})\phi_{ik}$ to the input rate r_k. Therefore, the contribution of the increment of r_i to the power consumption of node l can be expressed via r_k as $\epsilon(1-q_{ik})\phi_{ik}\partial\tilde{w}_l/\partial r_k$. Since this analysis is applicable for all next-hop neighbors, summing up over all $k \in S_i$ gives

$$\frac{\partial\tilde{w}_l}{\partial r_i} = \sum_{k\in S_i}(1 - q_{ik})\phi_{ik}\frac{\partial\tilde{w}_l}{\partial r_k} \tag{11}$$

Now suppose that the traffic λ_i is fixed, an increment ϵ to the routing variable ϕ_{ik} will cause an increment $\epsilon(1-q_{ik})\lambda_i$ to node k, which is equivalent to an increment of $\epsilon\lambda_i(1 - q_{ik})$ to input rate r_k. Applying the similar analysis as above, we find

$$\frac{\partial\tilde{w}_l}{\partial\phi_{ik}} = \lambda_i(1 - q_{ik})\frac{\partial\tilde{w}_l}{\partial r_k} \tag{12}$$

For example, in Fig. 1(a), node 1 is the source node and node 4 is the bottleneck node. An increment ϵ of the input rate of node 1 leads to an increment $\epsilon\phi_{12}$ to the traffic of node 2, which is equivalent to an increment of $\epsilon(1 - q_{12})\phi_{12}$ to the input rate of node 2. Similar analysis is applicable for node 3. So the overall increment of power consumption of node 4 due to the increment of r_1 is given by $\epsilon\partial\tilde{w}_4/\partial r_1 = \epsilon[(1 - q_{12})\phi_{12}\partial\tilde{w}_4/\partial r_2$

$+(1-q_{13})\phi_{13}\partial\tilde{w}_4/\partial r_3]$. Canceling out ϵ gives the result as (11). Similarly, an increment ϵ to the routing variable ϕ_{12} gives rise to an equivalent increment of $\epsilon\lambda_1(1-q_{12})$ to r_2, so the corresponding increment of power consumption of node 4 is expressed by $\partial\tilde{w}_4/\partial\phi_{12} = \lambda_1(1-q_{12})\partial\tilde{w}_4/\partial r_2$.

(b) **Adjacent source node**

If the source node i is adjacent to the bottleneck node l, in this case, the increment of power consumption of node l due to the increment of the input rate r_i consists of two parts. One is for receiving the increased traffic $\epsilon\phi_{il}$, which is given by $\epsilon\phi_{il}(e_r/E_l)$. The other is for transmitting the traffic $\epsilon(1-q_{il})\phi_{il}$, which is given by $\epsilon(1-q_{il})\partial\tilde{w}_l/\partial r_l$ following the similar analysis as above. Taking into account the indirect increment from other neighbor $k \neq l$, which can derived as above, we obtain

$$\begin{aligned}\frac{\partial\tilde{w}_l}{\partial r_i} &= \sum_{k\in S_i, k\neq l}(1-q_{ik})\phi_{ik}\frac{\partial\tilde{w}_l}{\partial r_k} + \ \phi_{il}\left(\frac{e_r}{E_l} + (1-q_{il})\frac{\partial\tilde{w}_l}{\partial r_l}\right)\\ &= \sum_{k\in S_i}(1-q_{ik})\phi_{ik}\frac{\partial\tilde{w}_l}{\partial r_k} + \frac{\phi_{il}e_r}{E_l}\end{aligned} \tag{13}$$

Similarly, an increment ϵ to ϕ_{ik} leads to an increment of $\epsilon\lambda_i$ to node k, therefore

$$\frac{\partial\tilde{w}_l}{\partial\phi_{ik}} = \lambda_i\left(\frac{e_r}{E_l} + (1-q_{il})\frac{\partial\tilde{w}_l}{\partial r_l}\right) \tag{14}$$

An example is illustrated in Fig. 1(b) where node 2 is the source node and node 4 is the bottleneck node. The increment ϵ of the input rate r_2 leads to an increment of $\epsilon\phi_{24}$ to node 4. The increment of the power consumption of node 4 is therefore given by $\epsilon\phi_{24}e_r/E_4$ plus $\epsilon(1-q_{24})\phi_{24}\partial\tilde{w}_4/\partial r_4$. Taking into account the increment from other downstream links gives rise to the result of (13). Similarly, the increment of power consumption of node 4 due to the increment ϵ of the routing variable ϕ_{24} is given by $\partial\tilde{w}_4/\partial\phi_{24} = \lambda_2(e_r/E_4 + (1-q_{il})\partial\tilde{w}_4/\partial r_2)$.

(c) **Co-located source node**

If the source node i is also a bottleneck node, note that r_i is a dummy variable, so we do not consider the power consumption for generating traffic ϵ. Taking the derivative directly from (5) we have

$$\frac{\partial\tilde{w}_i}{\partial r_i} = \sum_{k\in S_i}\frac{e_{ik}\phi_{ik}}{E_i} \tag{15}$$

$$\frac{\partial\tilde{w}_i}{\partial\phi_{ik}} = \frac{\lambda_i e_{ik}}{E_i} \tag{16}$$

The corresponding example is illustrated in Fig. 1(c) where source node 4 is also bottleneck node. The analysis is simple and we will not elaborate here.

Another possible case is for $i \in N_C$ and $l \in N_B$. However, this case is not necessary to discuss because both $\partial\tilde{w}_l/\partial r_i$ and $\partial\tilde{w}_l/\partial\phi_{ik}$ are zeros as $\tilde{w}_l$ has no relation with r_i.

We can now combine the above results to derive $\partial U/\partial \phi_{ik}$ of (10) by considering the following four cases:

- If $i, k \in N_A$, then none of the bottleneck nodes are adjacent to node i, so we can obtain $\partial w_l/\partial \phi_{ik}$ from (12) for all $l \in N_B$. Substituting these into (10) we have
$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2\lambda_i(1-q_{ik})}{|N_B|} \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \frac{\partial \tilde{w}_l}{\partial r_k} \tag{17}$$
- If $i \in N_A, k \in N_B$, then node k is a bottleneck node adjacent to node i. Therefore, $\partial w_k/\partial \phi_{ik}$ is given by (14), while for other bottleneck nodes $l \neq k$, $\partial w_l/\partial \phi_{ik}$ is given by (12). Substituting these into (10), we have
$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2\lambda_i}{|N_B|} \left((1-q_{ik}) \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \frac{\partial \tilde{w}_l}{\partial r_k} + \frac{e_r}{E_k} [c\tilde{w}_k - (c-1)\mu] \right) \tag{18}$$
- If $i, k \in N_B$, then node i and k are adjacent bottleneck nodes, so $\partial w_i/\partial \phi_{ik}$ and $\partial w_k/\partial \phi_{ik}$ are given by (16) and (14) respectively. Therefore
$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2\lambda_i}{|N_B|} \left((1-q_{ik}) \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \frac{\partial \tilde{w}_l}{\partial r_k} + \frac{e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] + \frac{e_r}{E_k} [c\tilde{w}_k - (c-1)\mu] \right) \tag{19}$$
- If $i \in N_B, k \in N_C$, the source node i is also a bottleneck node, so $\partial w_i/\partial \phi_{ik}$ is given by (16), therefore
$$\frac{\partial U}{\partial \phi_{ik}} = \frac{2\lambda_i}{|N_B|} \left((1-q_{ik}) \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \frac{\partial \tilde{w}_l}{\partial r_k} + \frac{e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] \right) \tag{20}$$

Now all that are required is to find a stationary point for the routing variable $\boldsymbol{\phi}$ to minimize U. We summarize it as the necessary condition in the following theorem.

Theorem 1. *(*Necessary Condition*) Let $\partial U/\partial \phi_{ik}$ given by* (17)-(20)*, the necessary condition for a minimum of U with respect to $\boldsymbol{\phi}^*$ for all $i \in N_A \cup N_B, k \in S_i$ is*
$$\frac{\partial U}{\partial \phi^*_{ik}} = \begin{cases} = \nu_i, \ \phi^*_{ik} > 0; \\ \geq \nu_i, \ \phi^*_{ik} = 0. \end{cases} \tag{21}$$

Proof. Let us define the following Lagrange function
$$U(\boldsymbol{\phi}, \boldsymbol{\nu}, \boldsymbol{\mu}) = U + \sum_{i \in N} \nu_i \left(1 - \sum_{k \in S_i} \phi_{ik} \right) - \sum_{i \in N, k \in S_i} \mu_{ik} \phi_{ik} \tag{22}$$

Where $\boldsymbol{\nu} = (\nu_1, \cdots, \nu_N)$ and $\boldsymbol{\mu} = \{\mu_{ik}\}$ are the Lagrange multipliers. According to Kuhn-Tucker theorem, the necessary condition for a ϕ^* to be a local minimum for $U(\boldsymbol{\phi}, \boldsymbol{\nu}, \boldsymbol{\mu})$ is that there exist Lagrange multipliers $\nu_i^*, i \in N$ and $\mu_{ik}^*, i \in N, k \in S_i$ such that

$$\frac{\partial U}{\partial \phi_{ik}^*} - \nu_i^* - \mu_{ik}^* = 0 \tag{23}$$

$$\mu_{ik}^* = 0 \text{ , if } \phi_{ik}^* > 0, \forall i, k; \tag{24}$$

$$\mu_{ik}^* > 0 \text{ , if } \phi_{ik}^* = 0, \forall i, k. \tag{25}$$

Rearranging (23) to $\partial U / \partial \phi_{ik}^* = \nu_i^* + \mu_{ik}^*$, and taking into accounts of (24) and (25) will complete the proof of (21).

The necessary condition (21) states that all links (i, k) for which $\phi_{ik} > 0$ must have the same value of $\partial U / \partial \phi_{ik}$, and this value must be less than or equal to the value of $\partial U / \partial \phi_{ik}$ for the links on which $\phi_{ik} = 0$. However, as illustrated in [21], the condition (21) is not sufficient to minimize U because it is automatically satisfied if the traffic rate λ_i is zero, even though the routing can still be improved. To overcome this problem, we prove next that after removing the factor λ_i from (17)-(20), the sufficient condition to minimize U with respect to $\boldsymbol{\phi}$ for all $i \in N_A \cup N_B, k \in S_i$ is given by the following theorem.

Theorem 2. *(*Sufficient Condition*) Let $\partial U / \partial \phi_{ik}$ given by* (17)-(20)*, and define $\partial U / \partial r_k = \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \partial \tilde{w}_l / \partial r_k$, it is sufficient for a $\boldsymbol{\phi}^*$ to be a minimizer of U if for all $i \in N_A \cup N_B, k \in S_i$, there is*

$$(1 - q_{ik}) \frac{\partial U}{\partial r_k} \geq \frac{\partial U}{\partial r_i} \tag{26a}$$

$$(1 - q_{ik}) \frac{\partial U}{\partial r_k} + \frac{e_r}{E_k} [c\tilde{w}_k - (c-1)\mu] \geq \frac{\partial U}{\partial r_i} \tag{26b}$$

$$\begin{aligned}(1 - q_{ik}) \frac{\partial U}{\partial r_k} &+ \frac{e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] \\ &+ \frac{e_r}{E_k} (c\tilde{w}_k - (c-1)\mu) \geq \frac{\partial U}{\partial r_i}\end{aligned} \tag{26c}$$

$$(1 - q_{ik}) \frac{\partial U}{\partial r_k} + \frac{e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] \geq \frac{\partial U}{\partial r_i} \tag{26d}$$

where (26a)-(26d) *correspond to the four cases given by* (17)-(20) *respectively.*

Proof. Suppose that there is a set of routing variables $\boldsymbol{\phi}^*$ satisfying (26), the corresponding node flows are $\boldsymbol{\lambda}^*$ and link flows are $\boldsymbol{f}^*$, where $f_{ik} = \lambda_i \phi_{ik}, i \in N, k \in S_i$. Let $\boldsymbol{\phi}$ be any other set of routing variables with the corresponding node flows $\boldsymbol{\lambda}$ and link flows $\boldsymbol{f}$. Define $f(\theta)$ as the convex combination of $\boldsymbol{f}^*$ and $\boldsymbol{f}$ with respect to a variable θ, that is,

$$f_{ik}(\theta) = (1 - \theta) f_{ik}^* + \theta f_{ik} \tag{27}$$

Therefore, each $\tilde{w}_l, l \in N_B$ can be represented by the link flow $\boldsymbol{f}$, which in turn is a function of θ, so U is also a function of θ. We rewrite the smoothing function (9) as

$$U(\theta) = \frac{c}{|N_B|} \sum_{l \in N_B} \tilde{w}_l^2(\theta) - \frac{(c-1)}{|N_B|^2} \left(\sum_{l \in N_B} \tilde{w}_l(\theta) \right)^2 \tag{28}$$

Since each $w_l(\theta)$ is a convex function of the node flow $\boldsymbol{f}$, therefore $U(\theta)$ is also a convex function with respect to θ, so it is obvious

$$\left. \frac{dU(\theta)}{d\theta} \right|_{\theta=0} \leq U(\boldsymbol{\phi}) - U(\boldsymbol{\phi}^*) \tag{29}$$

Since $\boldsymbol{\phi}$ is an arbitrary set of routing variable, it will complete the proof by proving that $dU(\theta)/d\theta \geq 0$ at $\theta = 0$.

From (5) and (27), it is straightforward to express $\tilde{w}_l$ as a function of the link flow $\boldsymbol{f}(\theta)$ as

$$\tilde{w}_l(\theta) = \frac{1}{E_l} \left(e_s + e_g r_l + \sum_{i \in A_l} f_{il}(\theta) e_r + \sum_{k \in S_l} f_{lk}(\theta) e_{lk} \right) \tag{30}$$

Differentiating $\tilde{w}_l$ directly from (27) and (30), we get

$$\frac{\partial \tilde{w}_l}{\partial \theta} = \sum_{i \in A_l} \frac{e_r}{E_l} (f_{il} - f_{il}^*) + \sum_{k \in S_l} \frac{e_{lk}}{E_l} (f_{lk} - f_{lk}^*) \tag{31}$$

We can calculate $dU(\theta)/d\theta$ directly using (28) and (31)

$$\begin{aligned}
\left. \frac{dU(\theta)}{d\theta} \right|_{\theta=0} &= \frac{2c}{|N_B|} \sum_{l \in N_B} \tilde{w}_l \frac{\partial \tilde{w}_l}{\partial \theta} - \frac{2(c-1)}{|N_B|^2} \sum_{l \in N_B} \tilde{w}_l \sum_{l \in N_B} \frac{\partial \tilde{w}_l}{\partial \theta} \\
&= \frac{2}{|N_B|} \sum_{l \in N_B} \left[c\tilde{w}_l - \frac{(c-1)}{|N_B|} \sum_{l \in N_B} \tilde{w}_l \right] \cdot \frac{\partial \tilde{w}_l}{\partial \theta} \\
&= \frac{2}{|N_B|} \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \cdot \left(\sum_{i \in A_l} \frac{e_r}{E_l} (f_{il} - f_{il}^*) + \sum_{k \in S_l} \frac{e_{lk}}{E_l} (f_{lk} - f_{lk}^*) \right)
\end{aligned}$$

We then first prove that

$$\sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \cdot \left(\sum_{i \in A_l} \frac{e_r f_{il}}{E_l} + \sum_{k \in S_l} \frac{e_{lk} f_{lk}}{E_l} \right) \geq \sum_{i \in N_A \cup N_B} r_i \frac{\partial U}{\partial r_i} \tag{32}$$

Note that from (26a)-(26d), multiplying both sides of these equations with λ_i and ϕ_{ik}, summing over all $i \in N_A \cup N_B$ and $k \in S_i$, and using the fact that $\lambda_i = r_i + \sum_{j \in A_i} \lambda_j \phi_{ji}(1 - q_{ji})$, we can obtain the result for the left-hand side as

$$\begin{aligned}\text{LHS} = &\sum_{i\in N_A\cup N_B}\sum_{k\in S_i}\lambda_i\phi_{ik}(1-q_{ik})\frac{\partial U}{\partial r_k} \\ &+\sum_{i\in N_A}\sum_{k\in S_i,k\in N_B}\left(\frac{\lambda_i\phi_{ik}e_r}{E_k}[c\tilde{w}_k-(c-1)\mu]\right) \\ &+\sum_{i\in N_B}\sum_{k\in S_i,k\in N_B}\left(\frac{\lambda_i\phi_{ik}e_{ik}}{E_i}[c\tilde{w}_i-(c-1)\mu]+\frac{\lambda_i\phi_{ik}e_r}{E_k}[c\tilde{w}_k-(c-1)\mu]\right) \\ &+\sum_{i\in N_B}\sum_{k\in S_i,k\in N_C}\left(\frac{\lambda_i\phi_{ik}e_{ik}}{E_i}[c\tilde{w}_i-(c-1)\mu]\right)\end{aligned} \tag{33}$$

and the right-hand side as

$$\text{RHS} = \sum_{i\in N_A\cup N_B} r_i\frac{\partial U}{\partial r_i}+\sum_{i\in N_A\cup N_B}\sum_{j\in A_i}\lambda_j\phi_{ji}(1-q_{ji})\frac{\partial U}{\partial r_i} \tag{34}$$

Now let look at the first term of LHS in (33), which sums over all links directed from nodes $i \in N_A \cup N_B$. Similarly, the second term of RHS in (34) sums over all in links directed to nodes $i \in N_A \cup N_B$. Recalling that the network is directed acyclic, canceling the common part of these two terms, the remaining part of the first term of (33) is the sum over all links $(i,k), i \in N_B, k \in N_C$, which is zero because $\partial\tilde{w}_i/\partial r_k$ are zero for these links. In other words, we can totally cancel out the first term of (33) and the second term of (34).

Re-arranging the summation of the second, third and the fourth terms of lefthand side in (33), and recalling the inequality between (33) and (34) , we obtain

$$\sum_{l\in N_B}[c\tilde{w}_l-(c-1)\mu]\left(\sum_{i\in A_l}\frac{e_r}{E_l}\lambda_i\phi_{il}+\sum_{k\in S_l}\frac{e_{lk}}{E_l}\lambda_l\phi_{lk}\right)\geq\sum_{i\in N_A\cup N_A} r_i\frac{\partial U}{\partial r_i} \tag{35}$$

Note that $f_{il} = \lambda_i\phi_{il}$, substituting this into (35) we can obtain (32).

Following the same derivation procedure, if λ^* and ϕ^* are substituted for λ and ϕ, this becomes an equality from the equations for $\partial U/\partial r_i$ in (26). That is,

$$\sum_{l\in N_B}[c\tilde{w}_l-(c-1)\mu]\left(\sum_{i\in A_l}\frac{e_r f^*_{il}}{E_l}+\sum_{k\in S_l}\frac{e_{lk}f^*_{lk}}{E_l}\right)=\sum_{i\in N_A\cup N_A} r_i\frac{\partial U}{\partial r_i} \tag{36}$$

Substituting (32) and (36) into (32), we see that $dW(\theta)/d\theta \geq 0$ at $\theta = 0$, which complete the proof.

3.3 Algorithm

Let us define two indicator functions I_i and I_k, where I_i is 1 if $i \in N_B$ and 0 otherwise, and I_k is 1 if $k \in N_B$ and 0 otherwise. Let $Z_{ik} = I_i e_{ik}[c\tilde{w}_i - (c-1)\mu]/E_i + I_k e_r[c\tilde{w}_k - (c-1)\mu]/E_k$, then the sufficient condition stated in (26) can be simplified as

$$(1-q_{ik})\frac{\partial U}{\partial r_k}+Z_{ik}\geq\frac{\partial U}{\partial r_i} \tag{37}$$

for all $i \in N_A \cup N_B, k \in S_i$, where equality is achieved for k whose routing variable ϕ_{ik} is greater than 0. That is, when the optimality is achieved, only those links with the smallest $(1 - q_{ik})\partial U/\partial r_k + Z_{ik}$ have nonzero traffic.

Based on the sufficient conditions, we design a gradient descent algorithm for each node to locally update its routing variables according to the received information from downstream neighbors. Instead of presenting the whole algorithm, we just present the routing variable update procedure here and refer the readers to standard textbooks such as [22] [23] for implantation details. Firstly, $(1 - q_{ik})\partial U/\partial r_k + Z_{ik}$ is computed for every neighbor $k \in S_i$. The best neighbor k_{min} with the smallest $(1 - q_{ik})\partial U/\partial r_k + Z_{ik}$ will have its routing variable increased while that of other neighbors' will be decreased accordingly. The next step is to compute the amount of reduction Δ_{ik} to each $\phi_{ik}(k \neq k_{min})$. Let a_{ik} be the gradient difference between each neighbor k and neighbor k_{min}, that is

$$a_{ik} = (1 - q_{ik})\frac{\partial U}{\partial r_k} + Z_{ik} - \min_{k \in S_i}\left\{(1 - q_{ik})\frac{\partial U}{\partial r_k} + Z_{ik}\right\},\ k \in S_i \tag{38}$$

Then the amount of traffic reduction Δ_{ik} is proportional to a_{ik} with the constraint that the routing variable ϕ_{ik} cannot be negative. That is, for each $k \in S_i, k \neq k_{min}$,

$$\Delta_{ik} = \min\left\{\phi_{ik}, \frac{\eta\phi_{ik}a_{ik}}{\max_{k \in S_i} a_{ik}}\right\} \tag{39}$$

and

$$\phi_{ik} \leftarrow \phi_{ik} - \Delta_{ik} \tag{40}$$

where η is a positive scalar. Finally, the total amounts of reduction are added to $\phi_{ik_{min}}$ as

$$\phi_{ik_{min}} \leftarrow \phi_{ik_{min}} + \sum_{k \in S_i, k \neq k_{min}} \Delta_{ik} \tag{41}$$

Using this algorithm, each node i gradually decreases the routing variables for which the value $(1 - q_{ik})\partial U/\partial r_k + Z_{ik}$ is large, and increases those for which it is small until the sufficient condition (37) is satisfied.

4 Performance Evaluation

We simulate the MLR algorithm over a set of sensor networks with the number of nodes varying from 50 to 100. Each network has its nodes randomly distributed over a square of 100 units by 100 units. All the nodes are assumed to have equal initial battery energy and equal traffic generating rate. For data correlation settings, we adopt the *gaussian random field* model [11] where the correlation coefficient q_{ik} decreases exponentially with the distance between nodes, or $q_{ik} = \exp(-\alpha d_{ik}^2)$. Here α is the correlation exponent and varies from $\alpha = 0.001$(high correlation) to $\alpha = 0.01$(low correlation) in the simulations. Also, a decreasing sequence of step size η and an increasing sequence of c are used in the simulations.

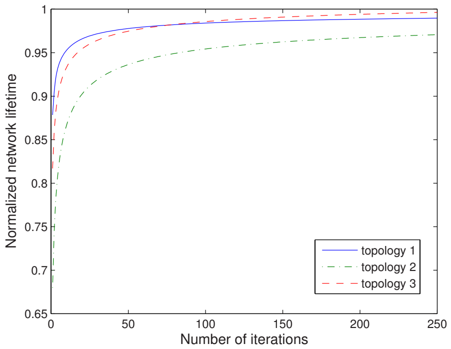


Fig. 2. Normalized network lifetime as a function of the number of iteration $(N = 50, \alpha = 0.005)$

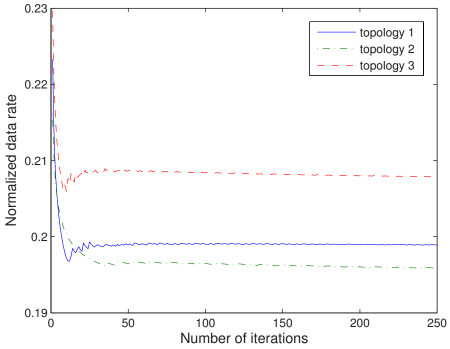


Fig. 3. Normalized data rate at sink node as a function of the number of iterations $(N = 50, \alpha = 0.005)$

Fig. 2 shows three traces of the network lifetime for three network topologies with 50 nodes. The network lifetime is computed at each iteration and normalized with respect to the optimal value obtained by the centralized solution to the MLR problem. We can see that the distributed algorithm can converge efficiently. For the same three sets of experiments, Fig. 3 shows the aggregated data rate at the sink node normalized to the total raw data rate of all source nodes. We observe that the traffic rate converges to a stable value in about 25 iterations, but from Fig. 2, we see that the network lifetime continue to increase after that. This is clearly due to the route optimization of the algorithm.

5 Conclusion

In this paper we have exploited the data correlation and optima routing to maximize the lifetime of a sensor network with a single sink node. We have proposed a smoothing function to overcome the nondifferentiability of the max function so that a distributed solution is possible. The optimality conditions are derived and a gradient decent algorithm is developed for every node to locally compute the routing variables. Simulation results show that the algorithm can converge to the optimal value efficiently and is scalable to the network size. Extension of our work for multiple sink nodes and for nodes with sleeping mode would be of interest, but these are beyond the scope of this paper.

Acknowledgment

This work is supported in part by the Hong Kong Research Grants Council under Grant CUHK 4220/03E.

References

1. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. International Conference on System Sciences*, 2000.
2. S. Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *MobiCom'98*, pages 181–190, 1998.
3. Teresa H. Meng V. Rodoplu. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, pages 1333–1344, August 1999.
4. J.-H Chang and L. Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. In *Proc. of Allerton Conference on Communication,Control and Computing*, Sep. 1999.
5. J.-H Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Infocom'00*, pages 22–31, 2000.
6. A. Sankar and Z. Liu. Maximum lifetime routing in wireless ad-hoc networks. In *Infocom'04*, March 2004.
7. R. Madan and S. Lall. Distributed algorithms for maximum lifetime routing in wireless sensor networks. In *Global Telecommunications Conference(GLOBECOM '04), IEEE*, volume 2, Nov 2004.
8. J. Pan, Y. Thomas Hou, L. Cai, Y. Shi, and Sherman X. Shen. Topology control for wireless sensor networks. In *MobiCom'03*, pages 286–299, 2003.
9. K. Kalpakis, K. Dasgupta, and P. Namjoshi. Maximum lifetime data gathering and aggregation in wireless sensor networks. In *Proc of ICN'02*, Aug. 2002.
10. Sundeep Pattem, Bhaskar Krishnamachari, and Ramesh. The impact of spatial correlation on routing with compression in wireless sensor networks. In *IPSN'04*, pages 28–35. ACM Press, 2004. Berkeley, California, USA.
11. R. Cristescu, B. Beferull-Lozano, and M. Vetterli. On network correlated data gathering. In *Infocom'04*, Hong Kong, 2004.
12. M. Gastpar and M. Vetterli. Source-channel communication in sensor networks. In *IPSN'03*, 2003.
13. Anna Scaglione and Sergio D. Servetto. On the interdependence of routing and data compression in multi-hop sensor networks. In *MobiCom '02*, pages 140–147. ACM Press, 2002. Atlanta, Georgia, USA.
14. Mehmet C. Vuran, Ozgur B. Akan, and Ian F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks*, 45(3):245, 2004.
15. F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *Proc. of DIALM'99*, pages 24–33, September 2002.
16. M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine*, 15(6):30–39, November 2001.
17. A. Ben-Tal and M. Teboulle. A smoothing technique for nondifferentiable optimization problems. In *Proceedings of the international seminar on Optimization*, pages 1–11, New York, NY, USA, 1988. Springer-Verlag New York, Inc.
18. X. Li. An entropy-based aggregate method for minimax optimization. *Engineering Optimization*, 18:277–285, 1992.
19. Chunhui Chen and O. L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Math. Program.*, 71(1):51–69, 1995.

20. L. Qi and D. Sun. Smoothing functions and a smoothing newton method for complementarity and variational inequality problems. *Journal of Optimization Theory and Applications*, 113:121–147, 2002.
21. Robert G. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transaction on Communications*, 25(1):73–85, Jan. 1977.
22. Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
23. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.

Managing Random Sensor Networks by means of Grid Emulation[⋆]

Zvi Lotker[1] and Alfredo Navarra[2]

[1] Centrum voor Wiskunde en Informatica Kruislaan 413,
NL-1098 SJ Amsterdam, Netherlands
lotker@cwi.nl
[2] Computer Science Department, University of L'Aquila.,
Via Vetoio I-67100 L'Aquila, Italy
navarra@di.univaq.it

Abstract. A common assumption in sensor networks is that the sensors are located according to a uniform random distribution. In this paper we show that uniform random points on the two dimensional unit square are almost a "grid". In particular, for a synchronous geographic sensor network we show how to emulate any grid protocol on random sensor networks, with high probability.

This suggests the following framework. In order to solve a problem on a random sensor network we solve the same problem on the grid. Then we use our emulation to make the obtained solution suitable for random sensor network. We analyze the cost of the emulation in terms of consumed energy and time. Finally we provide three examples that illustrate our method.

Keywords: Routing, Scheduling, MAC-layer, Collisions, Grid.

1 Introduction

A sensor network is usually modeled as a radio network where the sensors are spread out at random over a given area according to a uniform distribution. The structure of sensor networks is complex and presents many challenges. This is due to its random characteristic and its induced physical limitations (i.e., energy consumption, transmission range and open medium access constraints).

In a random sensor network usually each sensor does not have any knowledge about the network in which it is working, unless some local information is obtained by exchanging control messages with its neighbors. Moreover, since the sensors are placed at random, a first glance might suggest a total lack of structure. This is not necessarily the case. Dealing with randomness is always a problem. One way of dealing with it is by simulations. This solution is time and effort consuming and its accuracy is usually hard to evaluate. Another way of

[⋆] The research was partially funded by the European projects COST Action 293, "Graphs and Algorithms in Communication Networks" (GRAAL) and COST Action 295, "Dynamic Communication Networks" (DYNAMO).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 856–867, 2006.

approaching this problem is by applying sophisticated stochastic geometry tools. This approach is again time costly and it is not always simple. Understanding the structure of random sensor networks is a quintessential problem in the field of sensor networks. Clearly an understanding of this structure can lead to a major improvement in energy consumption and in the overall performance of the random network.

A standard and elegant technique when dealing with complex structures is to find a simpler structure that is close enough to the complex one, and yet simple enough to understand (see for instance [8]). This is our main goal in this work.

Our contribution is a grid protocol emulation for random sensor networks. In order to achieve this we develop optimal scheduling schemes that avoid collisions. More precisely we propose a general framework that is capable of emulating any protocol based on a grid structure for random sensors. In this way, we break the problem into 2 steps. The first step is to solve the problem on the grid. Since the grid is a well known and well researched structure, a textbook solution there probably already exists. The second step is to emulate the solution on the random sensor network using our grid emulation protocol. The advantages of this approach are evident. First, there are many problems that are already optimally solved on grids. Second, usually it is much easier to solve a problem on grids than on a random set of points. Moreover we are going to show that the cost of the grid emulation in terms of consumed energy and time is not too high. In particular, we use our method to solve the *Broadcast*, the *Gossiping* and the *Leafy Tree* problems on random sensor networks, obtaining satisfactory solutions. Last, the grid emulation can also be used as a rule of thumb to evaluate the correctness of simulations.

In order to achieve the grid emulation we develop a collision-free scheduling scheme. Using this scheduling scheme we developed a collision-free routing algorithm that can be easily applied in order to perform any desired communication on any sensed area of interest. Our scheme is completely independent of the routing protocol among the location-aware ones [1, 11]. It is worth noting that the combination of the routing protocol with the scheduling scheme is the main key for the conservation of energy in any communication. While the routing scheme, in fact, minimizes the energy needed to perform a desired communication, the scheduling prevents cases where communications must be repeated several times before succeeding. This concept was initiated by [7] where the authors dealt with random and deterministic scheduling functions. The main differences reside in their main assumptions for which each sensor is aware about the position of any other one and moreover each virtual grid square is assumed to be not empty. They also assume three basic states for the sensors. *Active*, when a sensor can transmit, *Passive*, when a sensor can receive and *Sleep* when a sensor is switched off in order to save energy. Concerning collisions, those are caused by superpositions of the transmission ranges of the sensors as in [3] but in [7] also by an extra range, called *interference range* (R_p). For the sake of clarity we do not cope with such an extra range but everything is easily scalable.

The paper is organized as follows. In the next section we describe the model and motivations that led to the assumptions made in the paper. In Section 3 we take care of the MAC-layer in order to avoid collisions in the communications. We also provide analysis in order to estimate the needed time for a source-destination communication. In Section 4 we show how the combination of a routing protocol with our scheduling scheme can be applied in order to emulate grid structures hence implying a virtual infrastructure on the network (see for instance [14]). Finally, in Section 5 we discuss some conclusive remarks.

2 Model

As assumed in the large majority of the papers we consider random instances of sensor networks in the two dimensional space (see [1, 11] for a survey on sensor networks routing protocols). The randomness of the spread sensors is usually motivated by the applications. The area of interest, in fact, where the sensing must be computed, can be an impervious, even dangerous area so that the sensors cannot be suitably set up. Without loss of generality we consider a square area using a uniform distribution. Each sensor knows its own location inside the considered area. Positioning information can be obtained through GPS systems, but also by cheaper means such as services like Ad-Hoc Positioning System (APS) [15] or the GPS-less low-cost outdoor localization for very small devices proposed in [4]. Sensors are assumed to be synchronized. As for the location awareness, the synchronization can be accomplished either by some strong assumption like a central clock to which each sensor refers (a GPS device can be also used for this purpose) or by means of cheaper strategies like the one presented in [16]. About the energy consumption concerning the sensor communications we refer to the most common power attenuation model [17] by which the signal power P_s of a sensor s decreases as a function of the distance in such a way that any station s' at distance $||s, s'||$ from s can receive a message from s if $P_s \geq O(||s, s'||^2)$. If a sensor is reached simultaneously by more than one transmission, a collision occurs and the received messages are assumed to be unreadable. Note that, in what follows, with "high probability" we mean a probability of $1 - \frac{1}{N}$ with $N = n \times n$ being the number of considered sensors.

3 MAC-Layer

In this section we describe a deterministic MAC-layer schedule based on the locations of the transmitters. For simplicity we assume that the sensors lie on a regular 2-dimensional grid G of $N = n \times n$ vertices V. We will remove this assumption in Section 4. For the sake of generality, we assume that some of the grid points are free from sensors and that some of them have more than one. The second case can be simplified just by considering one sensor in such grid points, since sensors in the same location can check the presence of overlapping ones without loosing too much energy and time. Moreover, we assume that each sensor knows its position but they do not know anything about the topology

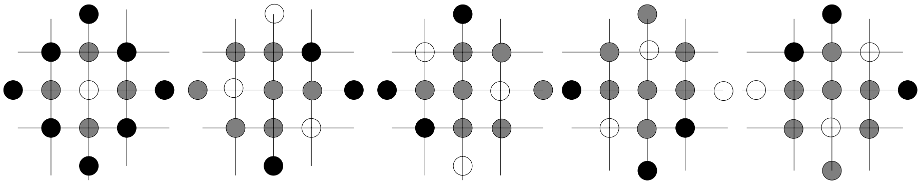


Fig. 1. Schedule scheme for 1-unit square grid transmissions. It needs 5 time slots to perform all the communications at distance 1. The white nodes are the transmitting one, the grey are the receivers and the black are inactive in order to avoid collisions.

of the network except that all the sensors are on some grid points. In order to save energy, collisions should be avoided. We now describe an algorithm to perform communications without collision. Since a sensor does not have information about the other sensors, we have to assign slots of communication to each pair of the network to ensure communication. A time slot is just a window of time during which some sensors are allowed to terminate one transmission operation. Its duration is dependent by the technology of the used sensors and without loss of generality we can consider one time slot as one unit of time (see for instance Figure 1).

Independently of the grid structure we need $\binom{N}{2} = \binom{n^2}{2}$ time slots (one for each possible pair). Indeed we can parallelize some of the transmissions in order to reduce the time needed to perform eventual communications.

Let $D = \{D(x,r) : x \in V, r \in \mathbb{R}\}$ be the set of disks of radius r centered at node x. A *schedule* $S : \mathbb{N} \to 2^D$ is a function from time step to a subset of disks. Next we define two properties of deterministic schedule.

Definition 1. *Let S be a deterministic schedule,*

1 *S has* no collisions *if any two nodes transmitting at the same time cannot reach a common node, i.e., $\forall c \in \mathbb{N}$, $S(c)$ is a subset of disjoint disks.*
2 *S is* universal *if any source $x \in V$ destination $y \in V$ pair x, y can communicate infinitely many times, i.e., $\forall x, y \in V$ and $t \in \mathbb{N}\, \exists t' > t : D(x,r) \in S(t')$ with $r \geq ||x-y||$.*

Let $S(x,y,k)$ be the number of slots in the schedule S that the node x needs to wait in order to communicate with node y for the k-th times.

Definition 2. *Let S be a schedule, the* fairness *of S is*

$$\phi(S) = \max_{x,y \in V, k \in \mathbb{N}} \{S(x,y,k) - S(x,y,k-1)\}.$$

Note that, without any information about the topology of the network, ϕ represents the time needed in the worst case to perform any communication.

Lemma 1. *For any universal schedule S without collisions $\phi(S) = \Theta(n^4)$.*

Proof. Assume, by contradiction, that $\phi(S) < \frac{n^4}{64}$. This means that considering any interval of time equal to $\frac{n^4}{64}$ we must find in S all the source-destination pairs. Since the number of pairs at distance more than $\frac{n}{2}$ is bigger than $\frac{n^4}{16}$ and that without collisions we can parallelize at most 4 of them, at least $\frac{n^4}{64}$ time slots are needed. The claim then holds by remembering that the number of all the source-destination pairs is $\binom{n^2}{2}$. □

Since we are interested in random points with uniform distribution, a natural question is whether we can improve the expectation of the communication time. Depending on the desired communications, in many cases a good idea for a routing algorithm may be to prefer short hop instead of long ones. This is due to the fact that short transmissions are less expensive in terms of consumed energy and moreover they can be parallelized much more than the long ones.

Let us divide all the source-destination pairs according to their Euclidean distance. Let $P = \{\pi_1, \pi_2, \ldots, \pi_d\}$ be such a partition where $\pi_i = \{(x, y) : x, y \in V(G)$ and $i - 1 \le ||x - y|| < i\}$ is the set of all the pairs at distance i on the grid and $d = \sqrt{2}n$ is the diameter of G. Following the previous ideas we want to perform all the communications of each π_i in the best way.

Since the disks close to the boundary of the grid are not full, we define $b(r) = \max_{x \in V(G)} |D(x, r)|$ to be the maximal number of grid points contained in a ball of radius r. Let opt_i be the optimal number of time slots needed to perform the communications defined by π_i.

Note that there is a big difference in the number of possible disjoint disks used by *opt* between the case of radius $r < \frac{n}{4}$ and the case of $r > \frac{n}{4}$ hence we describe two different procedures. In the first case, we consider a dense maximal disjoint packing $P_1(r)$ of the grid points by disks of the radius r. Since such a packing leaves holes between the circles, we need another shifted one, $P_2(r)$ to cover them (see for instance Figure 2).

Since for each disk in $P_1(r)$, (resp. $P_2(r)$) there are only 9 discs at distance less than $2r$ (see figure 3), we partition $P_1(r)$, (resp. $P_2(r)$) into 9 subparts in a way that all the distances between discs in each subpart is bigger than $2r$.

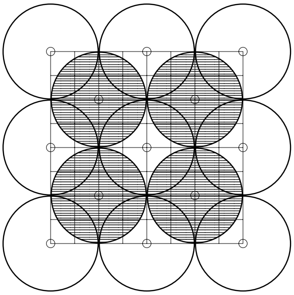


Fig. 2. The coverage of the whole grid by the two described complementary packings $P_1(r)$ (empty circles) and $P_2(r)$ (shaded circles)

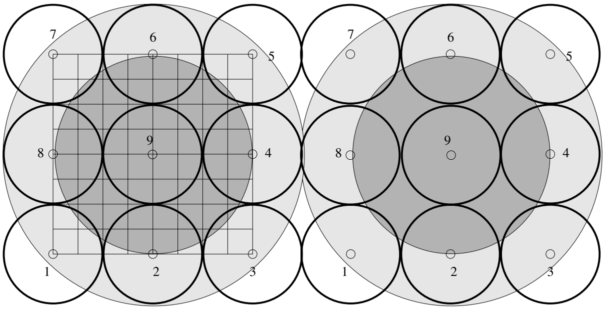


Fig. 3. The subpartition of P_1^j, $j = 1, ..., 9$. The numbers in the figure determine to which subpartition the node belongs. All the grey areas show the total area that can be covered by nodes from P_1^9. The dark grey areas show the nodes that receive the transmission from the central node in P_1^9. Note that in this case $b(3) = 13$, the total time it takes to get all the communications of π_3 is $13 \cdot 18 = 234$. For the sake of clarity the grid is shown just in the left part of the figure.

Denote P_i^j to be the $j = 1, ..., 9$ subparts of the packing $i = 1, 2$. We schedule the points covered by $P_1^j(r)$ to transmit before the ones covered only by $P_2^j(r)$.

Let $g_{0,0}$ be the point at the center of the Grid and let us consider the circle centered on it. We label the contained grid points from 1 to $b(r)$ in such a way that each node get a unique label. We use the same numbering process for each circle of both $P_1(r)$ and $P_2(r)$. This numbering represents the transmitting sequence in which every node of $P_1(r)$ (resp. $P_2(r)$) with the same label can simultaneously transmit.

```
procedure $\mathcal{S}(T, P_1(r), P_2(r))$
for $j = 1$ to 9 do
    Let $v_i$ be a node covered by $P_1^j$ and let $n_i$ be its label.
    for $i = 1$ to $b(r)$ do
        every node labelled as $n_i$ is allowed to transmit at radius $2r$ in the $(T + n_i + b(r)(j - 1))$-th time slot
    end for
    Let $v_i$ be a node covered by $P_2(r)$ and let $n_i$ be its label.
    for $i = 1$ to $b(r)$ do
        every node labelled as $n_i$ is allowed to transmit at radius $2r$ in the $(T + n_i + b(r)j)$-th time slot
    end for
end for
```

In the second case, that is, when $r > \frac{n}{4}$, if our schedule uses one disk in a time slot it is still ok since the optimal solution cannot parallelize too many of such communications, i.e., no more than 9. In this way we just loose a constant factor.

Lemma 2. *The schedule $\mathcal{S}(T, P_1(r), P_2(r))$ performs all the communications of π_i in $O(opt_i)$ time slots without any collision.*

Proof. Let us first provide a lower bound for opt_i in the case of $i \leq \frac{n}{4}$. Consider the disk $D(i)$ of radius i placed at $g_{0,0}$. Such a disk contains exactly $b(i)$ nodes. The disks centered in those points have an overlapping in $g_{0,0}$. This means that, in order to avoid the collision in the central node, all those nodes have to transmit in different time slots. Note that the Schedule algorithm $\mathcal{S}(T, P_1(r), P_2(r))$ needs $18b(i)$ time steps for all communications of π_i. This means that in this case we obtain a 18-approximation on the number of time slots needed to perform all the communications. If $i > \frac{n}{4}$ using packing arguments, opt_i cannot transmit with more than 9 disks at the same time, hence a 18-approximation holds. To see that $\mathcal{S}(T, P_1(r), P_2(r))$ is collision free we use the fact that the distance of the discs in each P_i^j is bigger than $2r$. Since each sensor transmits at radius $2r$, the sensors that transmit simultaneously do not interfere with each other, and the lemma follows. □

4 Grid Emulation

In this section we apply the previous results in order to achieve a general technique for emulating any grid protocol with random sensors. The idea is to "move" the points to a grid structure. The movements (Long or Short) are performed by increasing the radius of transmissions to ensure that all the neighbors of the grid structure can communicate. The difference between the Long and the Short movements concerns the size of the grid structure and the technique to calculate the relative locations of the points. More precisely, sometimes we use global or local information. Another important issue is the granularity of the considered grid. In what follows we also estimate the needed overhead for the consumed energy induced by our emulation strategy. Note that, since our scheduling scheme is placed at the MAC-layer, our results can be achieved with any location-aware routing protocol.

Let us assume a protocol $\mathcal{A}$ performed on grid networks. Actually for each node (x, y) of the grid a protocol $\mathcal{A}$ defines the instruction $\mathcal{A}_{x,y}(t)$ it has to compute at time t. Let Γ be a mapping from the set of random points P to the grid nodes. Note that Γ changes according to the size of the chosen grid. In order to perform the emulation we accumulate several time steps into one phase. Each phase can be considered as one basic time step in the protocol $\mathcal{A}$. The number of time steps that defines one phase is the output of the schedule we use in order to perform one single communication in the grid. Let μ be the maximal distance between any pair (x, y) and its image $\Gamma(x, y)$. From lemma 2 it follows that $\mathcal{S}(T, P_1(\mu + 1), P_2(\mu + 1))$ has no collision. Moreover the real distance between two sensors that are neighbors on the grid is less than or equal to $2\mu + 1$. It follows that two sensors that are neighbors on the grid can communicate with each other.

Long Movement. To achieve a one to one mapping between the grid points $G_{n,n}$ and the n^2 sensors we use the results of [18]. By allowing each sensor to move at most $O(\log^{\frac{3}{4}} n)$ we achieve such a matching Γ with high probability.

Therefore we have $\mu \leq O(\log^{\frac{3}{4}} n)$. Without loss of generality let $\mu \in \mathbb{N}$. The schedule will be $\mathcal{S}(T, P_1(\mu+1), P_2(\mu+1))$ according to the procedure described in Section 3. By Lemma 2, the time needed to perform it is then $O(\mu^2)$. Moreover, since it is possible that several (roughly $\log(n)$) sensors will be in the same grid square, we must multiply by a factor of $O(\log n)$ in order to enable all the possible communications given by the emulated protocol $\mathcal{A}$.

In this case we need global information to compute Γ, i.e., each sensor has to know its associated grid node. In order to perform every local communication round on the grid, using the scheduling algorithm of Section 3, we need time $\Theta(\log^{\frac{3}{2}} n)$ and also the energy must be multiplied by the same factor. This means that up to a poly-log factor we achieve an upper bound for the energy and time needed for random points in the plane to emulate the protocol $\mathcal{A}$. More precisely, using the long movement strategy we get the following theorem.

Theorem 1. *Any protocol $\mathcal{A}$ over a grid network $G_{n,n}$ can be emulated with high probability on a set of n^2 random points with stretch factors of $O(\log^{\frac{5}{2}} n)$ in time and $O(\log^{\frac{3}{2}} n)$ in energy.*

Note that, considering one source-destination pair, the previous method is the fastest one in terms of time (scheduling steps), on the other hand each sensor transmits for long distance, i.e., $O(\log(n)^{\frac{3}{4}})$. This is expansive in terms of energy consumption. This suggests to consider a suitable routing scheme in order to manage a good trade-off between the minimum delivery time and the minimum energy consumption. This remains a challenging issue according to the actual desired patterns of communication.

Short Movement. In this case we consider a grid $G_{O(\frac{n}{\sqrt{\log n}}),O(\frac{n}{\sqrt{\log n}})}$ but still with n^2 sensors, therefore there is still an average of $\log n$ nodes that belong to the same grid square. We associate to the left bottom grid node of each grid square one sensor lying in that square. This is accomplished by means of standard local leader election strategies [13], which costs $\log \log n$ time steps with high probability. We summarize the Short movement performance in the next theorem.

Theorem 2. *Any protocol $\mathcal{A}$ over a grid network $G_{\frac{n}{8\sqrt{\log n}},\frac{n}{8\sqrt{\log n}}}$ can be emulated with probability $1-\frac{1}{n^6}$ on a set of n^2 random points with constant stretch factors in time and in energy.*

*Proof (*sketch). In order to emulate $\mathcal{A}$ we need to have a sensor in each grid square. Note that the size of each square is $64 \log n$, and so the expected number of sensor in each grid square is $64 \log n$. Formally let $0 < i, j < \frac{n}{8\sqrt{\log n}}$. Denote the number of sensors in the grid square i, j by $X_{i,j}$. Using Chernoff we get

$$\Pr[X_{i,j} \leq 64(1-\lambda)\log(n)] \leq e^{-64\lambda^2 \frac{\log(n)}{2}}$$

Taking $\lambda = 1/2$ it follows that $\Pr[X_{i,j} \leq 32\log(n)] \leq \frac{1}{n^8}$. To bound the probability that none of the $\frac{n^2}{64\log n}$ grid squares in $G_{\frac{n}{8\sqrt{\log n}},\frac{n}{8\sqrt{\log n}}}$ is empty we use union bound. Since the number of grid squares is less than n^2 it follows that:

$$\Pr[Min\{X_{i,j} : 0 < i,j < \frac{n}{8\sqrt{\log n}}\} \leq 32\log(n)] < \frac{n^2}{n^8} = \frac{1}{n^6}$$

In order to perform every local communication round on the grid, using the scheduling algorithm of Section 3 it follows that we need a constant stretch factor in time and in energy. The constant time factor follows from the fact that we use π_1 scheduling. The energy constant follows from locality, i.e., two neighbors on the grid are at distance $8\sqrt{\log n}$ (on the grid), and their real distance on the plane is less than $\sqrt{3}8\sqrt{\log n}$. Moreover in the short movement, the grid neighborhood of each node coincides with the real neighborhood on the plane. Therefore using the π_1 scheduling we can perform all the desired communications. This is accomplished by the "trick" of throwing a number of sensors (n^2) that is bigger than the grid dimension ($\frac{n^2}{\log n}$). □

The Short Movement performs much better than the Long one both in time and in energy consumption. This is due to the fact that spreading more sensors than the number of grid nodes substantially increases the probability that some sensor is close to a grid node. In fact, each grid square may contain several points (usually $\log n$ points). Therefore, the short movement has an overhead generated in the initial stage due to the leader election which can be done in $O(\log\log(n))$. This leader will be the node that emulates the grid nodes. Moreover we can permute the leader role among all the sensors that belong to the same grid square. By doing this we balance the energy consumption among all the sensors, not only among the leaders. By doing this we prolong the lifetime of the network. The time needed to achieve this permutation is $O\log(n) \cdot \log\log(n))$. Since such a procedure is very local it is also not expensive in energy. By means of such movements we are finally able to remove the assumption of Section 3 for which the sensors were placed on the grid nodes.

4.1 Applications

In order to demonstrate the strength of our results, we now describe some important application problems on random sensor networks easily solvable by means of our technique with high probability related to the location of the sensors. We focus on the upper bounds obtained by such a method. Roughly speaking the idea is to consider a generic protocol $\mathcal{A}$ and perform it by the Long or the Short movement.

Broadcast. One of the most important protocols in any kind of communication network is given by the Broadcast protocol (see for instance [2, 5, 9, 10]). In [5] the authors describe the optimal algorithm for grid structure roughly showing that it needs $D + 2$ hops where D is the diameter of the grid. By applying the Short Movement we can achieve the Broadcast in $O(D)$ time and $O(n^2)$ energy hence solving the problem almost optimally. Note that, for the specific broadcast application it is useless to apply the Long Movement wasting much more time and energy.

Corollary 1. *$\mathcal{A} \equiv$ Broadcast over a grid network $G_{O(\frac{n}{\sqrt{\log n}}),O(\frac{n}{\sqrt{\log n}})}$ can be emulated with high probability on a set of n^2 random points with $O(\frac{n}{\sqrt{\log n}})$ time and $O(n^2)$ energy.*

Gossiping. Another important basic protocol in communication tasks is the gossiping. Each node participating in the protocol is assumed to have a value which should be transmitted to all the other ones. A trivial solution is then given by performing n^2 broadcasting communication, that is, one per node. In [19] a $O(n^3)$ deterministic Gossiping algorithm for radio networks of n^2 nodes is presented without any knowledge about the node locations. Restricting the attention to $G_{n,n}$ as shown in [6] the number of communications has an upper bound of $O(n^2)$ and the needed time has an upper bound of $O(n)$.

Corollary 2. *$\mathcal{A} \equiv$ Gossiping over a grid network $G_{n,n}$ can be emulated with high probability on a set of n^2 random points with $O(n \log^{\frac{5}{3}} n)$ time and $O(n^2$ $\log^{\frac{3}{2}} n)$ energy.*

In order to apply the Short Movement we have to pay attention to the values that belong to the nodes that are not actively participating in the protocol. We divide the protocol into two phases. In the first one each elected "leader", representative of every grid node, has to collect all the values belonging to the surrounding sensors that are physically associated to the same grid node (at most $O(\log n)$). This phase costs $O(\log n \log\log n)$. In the second phase the real gossiping starts between the grid nodes.

Corollary 3. *$\mathcal{A} \equiv$ Gossiping over a grid network $G_{O(\frac{n}{\sqrt{\log n}}),O(\frac{n}{\sqrt{\log n}})}$ can be emulated with high probability on a set of n^2 random points with $O(\frac{n}{\sqrt{\log n}})$ time and $O(\frac{n^2}{\log n})$ energy.*

Leafy Trees. Given a graph $G = (V, E)$ the problem is to find a spanning tree with a maximal number of leaves [12]. Such a problem is very interesting in the field of sensor networks since increasing the number of leaves reduces the number of needed transmissions and hops. Usually the underlying graph that models a sensor network is complete so the leafy tree can be trivially solved by one node connected to all the other ones hence obtaining $n^2 - 1$ leaves. Actually such a solution is practically not feasible due to the limited resources of the sensors, moreover, we are interested in the emulation of grid structures. On the full grid the maximal number of leaves is approximatively $\frac{2}{3}n^2$ (see Figure 4).

Using the Short Movement we obtain a number of leaves proportional to $\frac{2}{3}$ of the grid nodes plus all the nodes associated to the same grid node but one, hence obtaining,

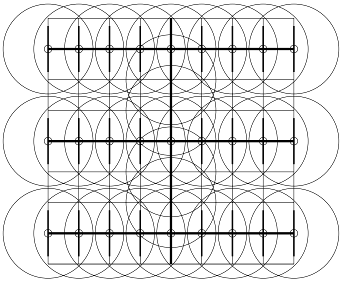


Fig. 4. The Leafy Tree for a grid network of 81 nodes. It contains 45 leaves.

Corollary 4. *$\mathcal{A} \equiv$ Leafy Tree over a grid network $G_{O(\frac{n}{\sqrt{\log n}}),O(\frac{n}{\sqrt{\log n}})}$ can be emulated with high probability on a set of n^2 random points obtaining roughly $\frac{2}{3}\frac{n^2}{\log n} + n^2(1 - \frac{1}{\log n})$ leaves.*

5 Conclusion

In this paper we have shown that the combination of routing and the MAC-layer can be efficient in a sensor network in terms of energy consumption and delivery time. We have proposed a scheduling scheme that perfectly matches with any location-aware routing protocol, hence obtaining a fully functional protocol for sensor networks. We have shown that a simple algorithm can avoid any collision when the sensors know their own location and when they are synchronized. Actually we have proposed a powerful framework able to emulate any protocol based on grid structures for random instances of sensors. This can be used as a rule of thumb, that is, instead of solving problems on random sensors, we solve the problems on grid networks and adapt the obtained solutions to the random instances. We have also shown the strength of the proposed framework by solving basic problems like *Broadcast*, *Gossiping* and *Leafy Trees*.

References

1. Al-Karaki, J. N., and Kamal, A. E. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications 11*, 6 (2004), 6–28.
2. Alon, N., Bar-Noy, A., Linial, N., and Peleg, D. A lower bound for radio broadcast. *Journal on Computer and System Sciences 43*, 2 (1991), 290–298.
3. Barriere, L., Fraigniaud, P., and Narayanan, L. Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *Proc. of the 5th Int. Workshop on Discrete algorithms and methods for mobile computing and communications (DIALM)* (2001), ACM Press, pp. 19–27.
4. Bulusu, N., Heidemann, J., and Estrin, D. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications 5* (2000).
5. Farley, A. M., and Hedetniem, S. T. Broadcasting in grid graphs. In *Proc. of the 9th S-E Conf. combinatorics, graph theory, and computing* (1978), pp. 275–288.

6. Farley, A. M., and Proskurowski, A. Gossiping in grid graphs. *Journal of Combinatorics, Information and System Science 5*, 2 (1980), 161–172.
7. Friedman, R., and Korland, G. Timed grid routing (tigr) bites off energy. In *Proc. of the 16th ACM Int. Symp. on Mobile ad hoc networking and computing (MobiHoc)* (2005), pp. 438–448.
8. Klasing, R., Lotker, Z., Navarra, A., and Perennes, S. From Balls and Bins to Points and Vertices. In *Proc. of the The 16th Annual Int. Symp. on Algorithms and Computation (ISAAC)* (2005), vol. 3827 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 757–766.
9. Kortsarz, G., and Peleg, D. Approximation algorithms for minimum time broadcast. In *Proc. of the Symp. on Theory of computing and systems (ISTCS)* (1992), Springer-Verlag, pp. 67–78.
10. Kowalski, D. R., and Pelc, A. Deterministic broadcasting time in radio networks of unknown topology. In *Proc. of the 43rd Symp. on Foundations of Computer Science (FOCS)* (2002), IEEE Computer Society, pp. 63–72.
11. Kozma, G., Lotker, Z., Sharir, M., and Stupp, G. Geometrically aware communication in random wireless networks. In *Proc. of the 23rd Annual ACM Symp. on Principles of distributed computing (PODC)* (2004), pp. 310–319.
12. Lu, H., and Ravi, R. The power of local optimization: Approximation algorithms for maximum-leaf spanning tree. In *Proc. of the 30th Annual Allerton Conf. on Communication, Control and Computing* (1992), pp. 533–542.
13. Malpani, N., Welch, J. L., and Vaidya, N. H. Leader election algorithms for mobile ad hoc networks. In *Proc. of ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)* (2000).
14. McCann, J. A., Navarra, A., and Papadopoulos, A. A. Connectionless Probabilistic (CoP) routing: an efficient protocol for Mobile Wireless Ad-Hoc Sensor Networks. In *Proc. of the 24th Int. Performance Computing and Communications Conf. (IPCCC)* (2005), pp. 73–77.
15. Niculescu, D., and Nath, B. Ad Hoc Positioning System (APS). In *Proc. of the 44th IEEE Global Telecommunications Conf. (GLOBECOM)* (2001).
16. PalChaudhuri, S., Saha, A. K., and Johnson, D. B. Adaptive clock synchronization in sensor networks. In *Proc. of the 3rd international Symp. on Information processing in sensor networks(IPSN)* (2004), ACM Press, pp. 340–348.
17. Rappaport, T. S. *Wireless communications: principles and practice.* Prentice-Hall, Englewood Cliffs, NY, 1996.
18. Shor, P. W., and Yukich, J. E. Minimax Grid Matching and Empirical Measures. *The Annals of Probability 19*, 3 (1991), 1338–1348.
19. Xu, Y. An $O(n^{1.5})$ deterministic gossiping algorithm for radio networks. *Algorithmica 36*, 1 (2003), 93–96.

Distributed Data Gathering in Multi-sink Sensor Networks with Correlated Sources

Kevin Yuen, Baochun Li, and Ben Liang

Department of Electrical and Computer Engineering,
University of Toronto, Ontario, Canada
{yuenke, bli}@eecg.toronto.edu, liang@comm.utoronto.ca

Abstract. In this paper, we propose an effective distributed algorithm to solve the minimum energy data gathering (MEDG) problem in sensor networks with multiple sinks. The problem objective is to find a rate allocation on the sensor nodes and a transmission structure on the network graph, such that the data collected by the sink nodes can reproduce the field of observation, and the total energy consumed by the sensor nodes is minimized. We formulate the problem as a linear optimization problem. The formulation exploits data correlation among the sensor nodes and considers the effect of wireless channel interference. We apply Lagrangian dualization technique on this formulation to obtain a subgradient algorithm for computing the optimal solution. The subgradient algorithm is asynchronous and amenable to fully distributed implementations, which corresponds to the decentralized nature of sensor networks.

Keywords: Sensor networks, data correlation, distributed algorithm, minimum energy, optimal rate allocation, transmission structure.

1 Introduction

Many applications for sensor networks, such as target tracking [1] and habitat monitoring [2], involve monitoring a remote or hostile field. Sensor nodes are assumed to be inaccessible after deployment for such applications and thus their batteries are irreplaceable. Moreover, due to the small size of sensor nodes, they carry limited battery power. Thus, energy is a scarce resource that must be conserved to the extent possible in sensor networks.

In this context, we are interested in solving the MEDG problem in multi-sink sensor networks with correlated sources. The first part of the problem objective is to find an optimal rate allocation on the sensor nodes, such that the aggregated data received by the sink nodes can be decoded to reproduce the entire field of observation. If the data collected by the sensor nodes are independent, then the rate allocation can be trivially determined – each sensor node can transmit at its data collection rate. However, sensor nodes are often densely deployed in sensor networks, hence the data collected by nearby sensor nodes are either redundant or correlated. This data correlation can be exploited to reduce the amount of data transmitted in the network, resulting in energy savings.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 868–879, 2006.

The second part of the problem objective is to find an optimal transmission structure on the network graph, such that the total energy consumed in transporting the data from the sensor nodes to the sink nodes is minimized. If the wireless links have unlimited bandwidth capacities, then each sensor node can transmit its collected data via the minimum energy path. However, as in any practical network, there are capacity limitations on the links and interference among competing signals. As a variation of wireless ad hoc networks, sensor networks have the unique characteristic of location-dependent contention. Signals generated by nearby sensor nodes will compete with each other if they access the wireless shared-medium at the same time. It is shown in [3] that the two parts of the problem objective can be achieved independently if capacity constraints do not exist. But in the presence of capacity constraints, the MEDG problem becomes complicated because the decision on the rate allocation will affect the decision on the transmission structure, and vice versa.

In this paper, we propose an efficient algorithm to solve the MEDG problem. The problem is carefully formulated as a linear optimization problem that can be solved with a distributed solution. This is important since centralized solutions require the participating nodes to repeatedly transmit status information across the network to a central computation node, thus they are not feasible for real-time calculations when energy constraints are present. To design a practical algorithm, we have assumed a realistic data correlation model and considered the effect of location-dependent contention. The formulation is relaxed with Lagrangian dualization technique and solved using the subgradient algorithm. The resulting algorithm is asynchronous, distributed, and supports large-scale sensor networks with multiple sink nodes.

Data gathering with correlated sources in sensor networks and resource allocation with capacity constraints in wireless networks have been separately studied in previous literature. The main contribution of this paper is to propose a solution to the MEDG problem that considers both topics simultaneously, and copes with the dependent relationship between the rate allocation and the transmission structure. To the best of our knowledge, no previous works have addressed the MEDG problem with all of the factors above.

2 Problem Formulation

2.1 Network Model

The wireless sensor network is modeled as a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of directed wireless links. Let S_N denote the set of sensor nodes and S_K denote the set of sink nodes. Then, $V = S_N \cup S_K$. The rate allocation assigns each sensor node $i \in S_N$ with R_i, which refers to a non-negative data collection rate. All sensor nodes have a fixed transmission range of r_{tx}. Let d_{ij} denote the distance between node i and node j. A directed link $(i, j) \in E$ exists if $d_{ij} \leq r_{tx}$. Each link is associated with a weight $e_{ij} = d_{ij}^2$, referring to the energy consumed per unit flow on link (i, j). All links are assumed to be symmetrical, where $e_{ij} = e_{ji}$. Moreover, f_{ij} represents the flow rate of link

(i, j). We have assumed that each sensor node has knowledge of its own location. Here, the rate vector $[R_i]_{\forall i \in S_N}$ and the flow vector $[f_{ij}]_{\forall (i,j) \in E}$ are the variables that can be adjusted in order to minimize the following optimization objective.

2.2 Optimization Objective

Given a rate allocation and a transmission structure, the flow rate on each link, denoted by f_{ij}, can be found and the total energy consumed on each link equals to $e_{ij} \cdot f_{ij}$. The objective of the MEDG problem is to minimize the total energy consumed in the network:

$$\textbf{Minimize} \quad \sum_{(i,j) \in E} e_{ij} \cdot f_{ij} \ . \tag{1}$$

2.3 Flow Conservation Constraints

For each sensor node $i \in S_N$, the total outgoing data flows must equal to the sum of the total incoming data flows and the non-negative data collection rate R_i. Since the sensor nodes relay all incoming data flows, only the sink nodes can absorb the data flows.

$$\sum_{j:(i,j) \in E} f_{ij} - \sum_{j:(j,i) \in E} f_{ji} = R_i, \quad \forall i \in S_N \ . \tag{2}$$

2.4 Channel Contention Constraints

The channel contention constraints model the location-dependent contention among the competing data flows. We build the constraints based on the protocol model [4] of packet transmission. According to the protocol model, all links originating from node k will interfere with link (i, j) if $d_{kj} < (1 + \triangle) d_{ij}$, where the quantity $\triangle > 0$ specifies a guard zone. We derive Ψ_{ij} for each link $(i, j) \in E$ as the cluster of links that cannot transmit as long as link (i, j) is active. The notation of cluster is used here as a basic resource unit, as compared to individual links in the traditional wireline networks. In sensor networks, the capacity of a wireless link is interrelated with other wireless links in its cluster. Therefore, data flows compete for the capacity of individual clusters, which is equivalent to the capacity of the wireless shared-medium. A flow vector $[f_{ij}]_{\forall (i,j) \in E}$ is supported by the wireless shared-medium if the channel contention constraints below hold:

$$f_{ij} + \sum_{(p,q) \in \Psi_{ij}} f_{pq} \leq C, \quad \forall (i, j) \in E \ , \tag{3}$$

where C is defined as the maximum rate supported by the wireless shared-medium. Note that the channel contention constraints are generic, since they can accommodate other models of packet transmission instead of the protocol model.

2.5 Rate Admissibility Constraints

Slepian-Wolf coding is introduced in [5]. It is an important work in exploiting data correlation among correlated sources. With Slepian-Wolf coding, sensor nodes are assumed to have correlation information of the entire network, and they can encode their data with only independent information. The Slepian-Wolf region specifies the minimum encoding rate that the sensor nodes must meet in order to transmit all independent information to the sink nodes. It is satisfied when any subset of sensor nodes encode their collected data at a total rate exceeding their joint entropy. In mathematical terms:

$$\sum_{i \in \mathbf{Y}} R_i \geq H(\mathbf{Y}|\mathbf{Y}^C), \quad \mathbf{Y} \subseteq S_N \ . \tag{4}$$

The rate admissibility constraints are non-linear since they grow at an exponential rate in relation to the number of nodes.

Since non-linear constraints are generally difficult to solve, it is desirable to remove them from the formulation. Moreover, the rate admissibility constraints require each sensor node to have global correlation information, which is not scalable in large networks. In this paper, we adapt a localized version of Slepian-Wolf coding from [6] to relax the rate admissibility constraints, such that only local correlation information is required at each sensor node. Here, we describe the localized Slepian-Wolf coding:

- Define a neighbourhood for each sensor node.
- Find the nearest sink node for each sensor node using a distributed shortest path algorithm, such as the Bellman-Ford algorithm [7]. Each sensor node refers to its nearest sink node as the destination sink node.
- For each sensor node i:
  - Find within its neighbourhood, the set N_i of sensor nodes that have the same destination sink node as node i, and are closer to that destination sink node than node i.
  - The Slepian-Wolf region is satisfied when node i transmits at rate $R_i = H(i|N_i)$.

Instead of global correlation information, the localized Slepian-Wolf coding only considers the correlation that a node has with its neighbourhood members. Based on a spatial data correlation model, it is natural to assume the nodes that are not in the neighbourhood contribute very little or nothing to the amount of compression. With a sufficient neighbourhood size, the localized coding should have a performance similar to global Slepian-Wolf coding. In this paper, we include the one-hop neighbours of the sensor nodes in their neighbourhoods.

2.6 Linear Programming Formulation

Combining the optimization objective with the introduced constraints, the MEDG problem can be modeled as a linear programming formulation.

$$\textbf{Minimize} \sum_{(i,j) \in E} e_{ij} \cdot f_{ij} \ , \tag{5}$$

$$\textbf{Subject to:} \quad \sum_{j:(i,j)\in E} f_{ij} - \sum_{j:(j,i)\in E} f_{ji} = H(i|N_i), \quad \forall i \in S_N \ , \tag{6}$$

$$f_{ij} + \sum_{(p,q)\in \Psi_{ij}} f_{pq} \le C, \quad \forall (i,j) \in E \ , \tag{7}$$

$$f_{ij} \ge 0, \quad \forall (i,j) \in E \ . \tag{8}$$

3 Distributed Solution: The Subgradient Algorithm

3.1 Lagrangian Dualization

The MEDG formulation resembles a resource allocation problem, where the objective is to allocate the limited capacities of the clusters to the data flows originating from the sensor nodes. Previous research works in wireline networks [8, 9] have shown that price-based strategy is an efficient mean to arbitrate resource allocation. In this strategy, each link is treated as a basic resource unit. A shadow price is associated with each link to reflect the traffic load of the link and its capacity. Based on the notation of maximal cliques, Xue *et al.* [10] extend the price-based resource allocation framework to respect the unique characteristic of location-dependent contention in wireless networks. Due to the complexities in constructing maximal cliques, the notation of cluster as defined in Section 2 is used as the basic resource unit. Each cluster is associated with a shadow price, and the transmission structure is determined in response to the price signals, such that the aggregated price paid by the data flows is minimized. It is revealed from previous research that at equilibrium, such price-based strategy can achieve global optimum.

To solve the MEDG formulation with a price-based strategy, we relax the channel contention constraints (3) with Lagrangian dualization technique to obtain the Lagrangian dual problem:

$$\textbf{Maximize} \ \mathrm{LS}(\beta) \ , \quad \textbf{s.t.} \quad \beta \ge 0 \ . \tag{9}$$

By associating price signals or Lagrangian multipliers β_{ij} with the channel contention constraints, the Lagrangian dual problem is evaluated via the Lagrangian subproblem LS(β):

$$\textbf{Minimize} \quad \sum_{(i,j)\in E} e_{ij} \cdot f_{ij} + \beta_{ij} \cdot (f_{ij} + \sum_{(p,q)\in \Psi_{ij}} f_{pq} - C) \ , \tag{10}$$

$$\textbf{Subject to:} \quad \sum_{j:(i,j)\in E} f_{ij} - \sum_{j:(j,i)\in E} f_{ji} = H(i|N_i), \quad \forall i \in S_N \ , \tag{11}$$

$$f_{ij} \ge 0, \quad \forall (i,j) \in E \ . \tag{12}$$

We further define Φ_{ij} as the set of clusters that link (i,j) belongs to. Recall Ψ_{pq} is the cluster of links that cannot transmit when link (p,q) is active. For any link (i,j) that interferes with link (p,q), link (i,j) belongs to the cluster of link (p,q).

Thus, for any links (i, j) and (p, q), $(p, q) \in \Phi_{ij}$ iff $(i, j) \in \Psi_{pq}$. The Lagrangian subproblem can be remodelled using this notation:

$$\textbf{Minimize} \quad \sum_{(i,j)\in E} f_{ij}(e_{ij} + \beta_{ij} + \sum_{(p,q)\in\Phi_{ij}} \beta_{pq}) - \beta_{ij}C \ , \qquad (13)$$

$$\textbf{Subject to:} \quad \sum_{j:(i,j)\in E} f_{ij} - \sum_{j:(j,i)\in E} f_{ji} = H(i|N_i), \quad \forall i \in S_N \ , \qquad (14)$$

$$f_{ij} \geq 0, \quad \forall (i,j) \in E \ . \qquad (15)$$

The objective function of the remodelled Lagrangian subproblem specifies that the weight of each link is equal to the sum of its energy and capacity cost. And the capacity cost is equal to the Lagrangian multiplier of the link plus the sum of the Lagrangian multipliers in Φ_{ij}. This is intuitive since when link (i, j) is active, any links in the set Φ_{ij} cannot transmit due to interference. So the actual price to pay for accessing link (i, j) should equal to the total price for accessing link (i, j) and all links in Φ_{ij}.

Since the capacity constraints are relaxed, we observe that the solution of the remodelled Lagrangian subproblem requires each sensor node to transmit its data along the shortest path that leads to its nearest sink node. As a result, the Lagrangian subproblem can be solved with a distributed shortest path algorithm, such as the Bellman-Ford algorithm [7]. Recall from the localized Slepian-Wolf coding scheme, a sensor node will co-encode with another sensor node only if they have the identical nearest sink node. Consequently, for any solution generated by the Lagrangian subproblem, data flows due to sensor nodes that have co-encoded with each other will be absorbed by an identical sink node.

3.2 Subgradient Algorithm

Many algorithms have been proposed to solve optimization problems, such as simplex, ellipsoid and interior point methods. These algorithms are efficient in the sense that they can solve large instance of optimization problems in a few seconds. However, they have the disadvantage of being inherently centralized, which implies that they are not applicable for distributed deployment. In this subsection, we describe the subgradient algorithm, a distributed solution to the Lagrangian dual problem.

The algorithm starts with a set of initial non-negative Lagrangian multipliers $\beta_{ij}[0]$. In our simulations, we set $\beta_{ij}[0]$ to zeros, assuming no congestion in the network. During each iteration k, given current Lagrangian multiplier values $\beta_{ij}[k]$, the Lagrangian subproblem is solved. Using the new primal values $[f_{ij}[k]]_{\forall(i,j)\in E}$ obtained from the Lagrangian subproblem, we update the Lagrangian multipliers by:

$$\beta_{ij}[k+1] = \max(0, \beta_{ij}[k] + \theta[k](f_{ij}[k] + \sum_{(p,q)\in\Psi_{ij}} f_{pq}[k] - C)) \ , \qquad (16)$$

where θ is a prescribed sequence of step sizes. If the step sizes are too small, then the algorithm has a slow convergence speed. If the step sizes are too large, then

β_{ij} may oscillate around the optimal solution and the algorithm fails to converge. However, the convergence is guaranteed [11], when θ satisfies the conditions $\theta[k] \geq 0$, $\lim_{k\to\infty} \theta[k] = 0$, and $\sum_{k=1}^{\infty} \theta[k] = \infty$. In this paper, we use the sequence of step sizes, $\theta[k] = \frac{a}{b+ck}$, where a, b, and c are positive constants.

The subgradient algorithm is an efficient tool for solving the Lagrangian dual problem. However, it has the disadvantage that an optimal solution, or even a feasible solution to the primal problem (the linear MEDG formulation) may not be available. We adapt the primal recovery algorithm introduced by Sherali *et al.* [11] to recover the primal optimal solution f_{ij}^*. At iteration k of the subgradient algorithm, the primal recovery algorithm composes a primal feasible solution $f_{ij}^*[k]$ via the solutions generated by the Lagrangian subproblem:

$$f_{ij}^*[k] = \sum_{m=1}^{k} \lambda_m^k f_{ij}[m] \ , \tag{17}$$

where $\lambda_m^k = \frac{1}{k}$ are convex weights. In this paper, for each iteration, the Lagrangian subproblem generates a rate allocation and a transmission structure. The primal recovery algorithm specifies that the solution to the MEDG problem (the optimal rate allocation and transmission structure) should equal to a convex combination of the solutions that are generated by the Lagrangian subproblem. Note that since each solution generated by the Lagrangian subproblem satisfies the Slepian-Wolf region, the convex combination of the solutions also satisfies the Slepian-Wolf region. In the kth iteration, we can calculate $f_{ij}^*[k]$ by:

$$f_{ij}^*[k] = \frac{k-1}{k} f_{ij}^*[k-1] + \frac{1}{k} f_{ij}[k] \ . \tag{18}$$

3.3 Distributed MEDG Algorithm

We now present our distributed algorithm for the MEDG problem. Each directed link (i,j) is delegated to its sender node i, and all computations related to link (i,j) will be executed on node i.

1. Choose initial Lagrangian multiplier values $\beta_{ij}[0]$, $\forall (i,j) \in E$.
2. For the kth iteration, determine the weight of each link as $(e_{ij} + \beta_{ij}[k] + \sum_{(p,q)\in \Phi_{ij}} \beta_{pq}[k])$.
3. Compute the shortest path from each sensor node to its nearest sink node using the distributed Bellman-Ford algorithm. Sensor nodes refer to their nearest sink node as their destination sink node.
4. For each sensor node i, determine its rate allocation according to the localized Slepian-Wolf coding scheme introduced in Section 2.
5. Based on the rate allocation and the transmission structure obtained, compute $f_{ij}[k+1]$ and $f_{ij}^*[k+1]$, for all links $(i,j) \in E$.
6. Update Lagrangian multipliers $\beta_{ij}[k+1] = \max(0, \beta_{ij}[k] + \theta[k](f_{ij}[k] + \sum_{(p,q)\in \Psi_{ij}} f_{pq}[k] - C))$, where $\theta[k] = \frac{a}{(b+ck)}$, for all links $(i,j) \in E$.
7. For each link (i,j), send $\beta_{ij}[k+1]$ to all links in Ψ_{ij} and send $f_{ij}[k+1]$ to all links in Φ_{ij}.
8. Repeat steps 2 to 7 until convergence.

4 Performance Evaluation

4.1 Data Correlation Model

Since the sensor nodes are continuous and not discrete sources, the theoretical tool to analyze the problem is Rate Distortion Theory [12]. Let S be a vector of n samples of the measured random field returned by n sensor nodes. Let $\hat{S}$ be a representation of S, and $d(S, \hat{S})$ be a distortion measure. With the mean square error (MSE) as the distortion measure, i.e., $d(S, \hat{S}) = ||S - \hat{S}||^2$, and the constraint $E(||S - \hat{S}||^2) < D$, a Gaussian source is the worst case, since it requires the most bits to be represented when compared with other sources [13]. For the purpose of illustration, we let S be a spatially correlated Gaussian random vector $\sim N(\mu, \Sigma)$. In this case, the rate distortion function of S is

$$R(\Sigma, D) = \sum_{n=1}^{N} \frac{1}{2} \log \frac{\lambda_n}{D_n} \quad , \tag{19}$$

where $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_N$ are the ordered eigenvalues of the correlation matrix Σ and

$$\sum_{n=1}^{N} D_n = D \quad , \quad D_n = \begin{cases} K & \text{if } K < \lambda_n \ , \\ \lambda_n & \text{otherwise} \ , \end{cases} \tag{20}$$

and K is chosen such that $\sum_{n=1}^{N} \min(K, \lambda_n) = D$. In our analysis, we let $\Sigma_{ij} = W^{d_{ij}^2}$, where W is a correlation parameter that represents the amount of data correlation between spatial samples. W should be less than one such that Σ is a semi-positive definite matrix. Given any subset of nodes X and the distortion per node d, we can construct its correlation matrix Σ_X and approximate its entropy with its rate distortion function, $H(X) \approx R(\Sigma_X, d \cdot |X|)$.

4.2 Simulation Environments

We study the distributed MEDG algorithm in three different simulation environments. In the *independent* environment, we neglect the effect of data correlation by substituting Slepian-Wolf coding with an independent coding scheme. In the *synchronous* environment, the participating nodes simultaneously execute an iteration of the algorithm at every time step. Bounded communication delay is assumed where price and rate updates will arrive at their destinations before the next time step. The *asynchronous* environment is based on the partial asynchronism model [10], which assumes the existence of an integer B that bounds the time between consecutive updates. To implement this environment, each sensor node maintains a timer with a random integer value between 0 and B. The timer decreases itself by 1 at every time step. When the timer reaches 0, the sensor node executes an iteration of the algorithm before resetting the timer. In this environment, update messages may be delayed or out-of-date.

The distributed MEDG algorithm is implemented with the C++ programming language. For all experiments, the transmission and interference range are set to 30m and the capacity of the wireless shared-medium is set to 600 bits. Unless stated, the experiments are executed on a random topology with 100 nodes, the correlation parameter W and the per node distortion d are set to 0.99 and 0.0001, respectively.

4.3 Convergence Behaviour

We study the convergence behaviour of our algorithm under the *synchronous* environment. To this end, we generate five random sensor fields, ranging from 100 to 500 nodes in increments of 100 nodes, with 10% of the nodes randomly chosen as sink nodes. The sensor field with 100 nodes has an area of 100m × 100m. Other sensor fields are generated by scaling the area to maintain a constant node density. The convergence speed of the algorithm is shown in Fig. 1. The optimal value is taken as the convergence value of the algorithm. We observe that it takes about 220 iterations to converge to 99% optimality in a network with 100 nodes, and this number increases to about 360 for a network with 500 nodes. Due to the slow increase in the number of iterations, the scalability of our algorithm is not affected by the network size. In addition, we notice that the algorithm can achieve 90% optimality in about half the iterations required to achieve 99% optimality. Therefore, in practice, when it is not necessary to achieve the optimal solution, we can obtain a near-optimal solution in a much shorter time. This result illustrates that our distributed algorithm is efficient for real-time calculations.

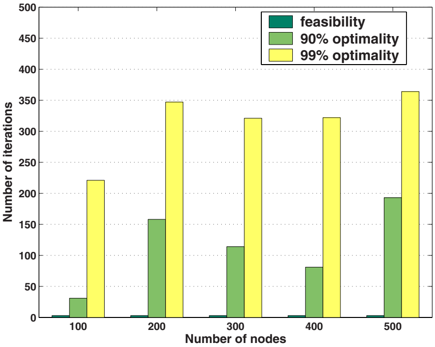


Fig. 1. Convergence speed of the distributed MEDG algorithm

4.4 Asynchronous Network Environments

To show that our algorithm is applicable in asynchronous network environments, we execute the algorithm under the *asynchronous* environment with different time bounds $B = 1, 2, 5, 10$. Each experiment is performed for 1000 time steps, and the total energy consumption attained at each time step is plotted in Fig. 2. In all four experiments, the algorithm converges to an identical optimal solution, which indicates that it can achieve convergence in asynchronous network environments. Moreover, we conclude that the convergence speed of the algorithm

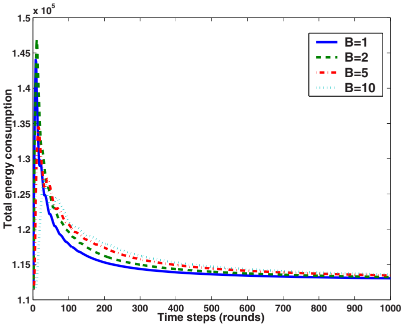


Fig. 2. Convergence in asynchronous network environments

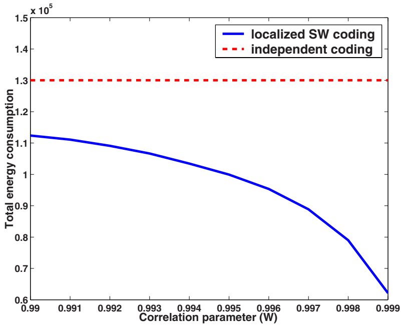


Fig. 3. Localized Slepian-Wolf coding vs. independent coding

is associated with the time bound B, since longer convergence time is required when B is large.

4.5 The Effect of Data Correlation

We investigate the effect of data correlation by comparing the *asynchronous* environment against the *independent* environment. As the correlation parameter W varies from 0.99 to 0.999, the total energy consumed by the different environments at convergence is recorded in Fig. 3. Clearly, the energy consumed at high correlation ($W = 0.999$) is much lower compared with the energy consumed at low correlation ($W = 0.99$). Overall, the localized Slepian-Wolf coding scheme outperforms the independent coding scheme by 15% to 50%. This result suggests that even though the algorithm utilizes only local information, it can achieve significant energy savings for a wide range of data correlation level.

5 Related Work

In [14], Kalpakis *et al.* have formulated the maximum lifetime data gathering and aggregation problem as an integer program. Although this formulation yields satisfactory performance, it makes the assumption of perfect data correlation, where intermediate sensor nodes can aggregate any number of incoming packets into a single packet. Perfect data correlation can also be found in [15], which analyzes the performance of data-centric routing schemes with in-network aggregation. We do not assume perfect data correlation in this paper since it may not be realistic in practical networks.

While our paper utilizes Slepian-Wolf coding, there are works that exploit data correlation with alternative techniques. Single-input coding is considered in [3, 16], where intermediate nodes can aggregate their collected data with the side information provided by another node. Cristescu *et al.* [3] prove that solving the MEDG problem with single-input coding is NP-hard, even in a simplified network setting. Since single-input coding can only exploit data correlation between pairs of nodes, it will not perform as well as Slepian-Wolf coding. In contrast, data aggregation with multi-input coding is performed when all input information from multiple nodes is available. Goel *et al.* [17] consider the joint treatment of data aggregation and transmission structure with multi-input coding. Although multi-input coding exploits data correlation among multiple nodes, it requires the nodes to explicitly communicate with each other. Since Slepian-Wolf coding does not require such communication, it can be implemented in asynchronous network environments without timing assumptions.

Other closely related works are the ones involving Slepian-Wolf coding. In [18], Servetto *et al.* introduced the sensor reachback problem, which requires a single node in the sensor network to receive sufficient data to reproduce the entire field of observation. Slepian-Wolf coding is employed to meet this requirement. This paper inspires us to apply Slepian-Wolf coding in the MEDG problem, allowing the sink nodes to receive independent information from all sensor nodes. In [6], Cristescu *et al.* address the MEDG problem with Slepian-Wolf coding. However, their optimization problem does not consider the effect of wireless channel interference, hence the solution generated may not be supported by the wireless shared-medium.

6 Conclusion

In this paper, we have presented an efficient solution for the MEDG problem in multi-sink sensor networks with correlated sources. The problem is carefully formulated as a linear optimization problem with a distributed solution. In the presence of capacity constraints, we show that finding the optimal rate allocation and transmission structure are two dependent problems, hence they must be addressed simultaneously. With a realistic model, the formulation exploits data correlation among the sensor nodes, accounts for location-dependent contention in the wireless shared-medium, and minimizes the total energy consumed by the network. Sensor nodes are required to transmit at a rate that satisfies the Slepian-Wolf region, which implies that the sink nodes will be able to reproduce the entire field monitored by the sensor network. The algorithm is amenable to fully distributed implementations, as the participating nodes are only needed to communicate with other nodes in their neighbourhood. The algorithm is asynchronous and provides multi-sink support, making it feasible for practical deployment in large-scale sensor networks. To the best of our knowledge, this is the first work that addresses the MEDG problem with data correlation and wireless channel interference simultaneously, especially when a price-based approach is employed to obtain a distributed solution.

References

1. Clouqueur, T., Phipatanasuphorn, V., Ramanathan, P., Saluja, K.K.: Sensor Deployment Strategy for Detection of Targets Traversing a Region. In: ACM Mobile Networks and Applications. Volume 8. (2003) 453–461

2. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D.: Wireless Sensor Networks for Habitat Monitoring. In: Proc. of First ACM International Workshop on Wireless Sensor Network and Applications. (2002)
3. Cristescu, R., Beferull-Lozano, B., Vetterli, M.: On Network Correlated Data Gathering. In: Proc. of IEEE INFOCOM. (2004)
4. Gupta, P., Kumar, P.R.: The Capacity of Wireless Networks. IEEE Trans. Information Theory **46**(2) (2000) 388–404
5. Slepian, D., Wolf, J.K.: Noiseless Coding of Correlated Information Sources. IEEE Trans. on Information Theory **4**(IT-19) (1973) 471–480
6. Cristescu, R., Beferull-Lozano, B., Vetterli, M.: Networked Slepian-Wolf: Theory and Algorithms. In: Proc. of European Workshop on Wireless Sensor Networks. (2004)
7. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Prentice Hall (1989)
8. Kelly, F.P., Maulloo, A.K., Tan, D.K.H.: Rate Control in Communication Networks: Shadow prices, Proportional Fairness and Stability. In: Journal of the Operational Research Society. Volume 49. (1998) 237–252
9. Low, S.H., Lapsley, D.E.: Optimization Flow Control: Basic Algorithm and Covergence. In: IEEE/ACM Trans. on Networking. Volume 7. (1999) 861–874
10. Xue, Y., Li, B., Nahrstedt, K.: Optimal Resource Allocation in Wireless Ad Hoc Networks: A Price-Based Approach. In: to appear in IEEE Transactions on Mobile Computing. (2005)
11. Sherali, H.D., Choi, G.: Recovery of primal solutions when using subgradient optimization methods to solve Lagrangian duals of linear programs. In: Operations Research Letter. Volume 19. (1996) 105–113
12. Cover, T.M., Thomas, J.A.: Elements of Information Theory. New York: Wiley (1991)
13. Lotfinezhad, M., Liang, B.: Effect of Partially Correlated Data on Clustering in Wireless Sensor Networks. In: Proc. of the IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON). (2004)
14. Kalpakis, K., Dasgupta, K., Namjoshi, P.: Efficient Algorithms for Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks. Computer Networks Journal (2002)
15. Krishnamachri, B., Estrin, D., Wicker, S.: Modelling Data-centric Routing in Wireless Sensor Networks. In: Proc. of IEEE INFOCOM. (2002)
16. Rickenbach, P.V., Wattenhofer, R.: Gathering Correlated Data in Sensor Networks. In: Proc. of DIALM-POMC '04: Proceedings of the 2004 joint workshop on Foundations of mobile computing. (2004) 60–66
17. Goel, A., Estrin, D.: Simultaneous Optimization for Concave Costs: Single Sink Aggregation or Single Source Buy-at-Bulk. In: Proc. of the 14^{th} Symposium on Discrete Algorithms (SODA). (2003)
18. Barros, J., Servetto, S.D.: Network Information Flow with Correlated Sources. Submitted to the IEEE Transactions on Information Theory, November 2003 (Original title: The Sensor Reachback Problem) Revised (2005)

Abstract Frames for Reducing Overhearing in Wireless Sensor Networks

Abdelmalik Bachir[1], Dominique Barthel[1],
Martin Heusse[2], and Andrzej Duda[2]

[1] France Telecom R&D, Meylan, France
[2] LSR-IMAG Laboratory, Grenoble, France

Abstract. We present a novel idea for energy saving by avoiding the reception of redundant broadcast frames. It is based on sending an abstract frame just before a data frame: the former contains a digest of the latter. We evaluate the energy savings of this scheme analytically and by means of simulation in ns-2. Although we have applied our approach to SMAC, the key idea is generic and can be used to reduce energy consumed by broadcast frames in a large variety of MAC protocols.

1 Introduction

Energy saving is crucial in designing long-lived sensor networks, mainly because nodes are powered by batteries that may be costly, difficult, or impossible to replace or to recharge. Measurements presented in the literature [1, 2] confirmed by our experiments with the MC 13192 SARD sensor node show that radio communication is a major source of energy consumption. Therefore, a node needs to switch the radio off whenever possible to save energy. At the same time, to be able communicate with other nodes, they need to activate their radios during some common active periods. Moreover, their competition for the radio channel should follow a set of rules defined in a MAC layer. Energy may be wasted due to the inherent behavior of the MAC layer because of the following problems:

- *Idle Listening*: since a node does not know when it will be the receiver of a frame, it must keep its radio in receive mode, which consumes energy.
- *Collisions*: the energy used to send and receive a frame is wasted when the frame collides with another frame.
- *Overhearing*: this happens when a sensor node receives and decodes an irrelevant frame (e.g. a broadcast frame that has already been received).
- *Protocol Overhead*: control frames do not carry useful information although their transmission consumes energy.

SMAC is an example of an energy efficient access method that tries to address all these issues [3].

We focus on the problem of reducing the impact of idle listening and overhearing. We observe that many applications require an one-to-all communication scheme such as a *network-wide broadcast*. A straightforward way to achieve this scheme is *flooding*: a node broadcast a message to its neighbors that forward it

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 880–891, 2006.

further on so that all the nodes eventually receive it. Network-wide broadcast is usually used in management protocols such as route discovery of on-demand routing protocols [4] or during the interest propagation phase in Directed Diffusion [5], and in many application protocols. For a given node in the network, only the broadcast frame received first is actually relevant and all subsequent ones are redundant since they carry the same data contents. Redundant transmissions degrade the performance of the network: when combined with a contention based MAC protocol, redundant transmissions increase the collision rate as reported in the broadcast storm problem [6]. Moreover, redundant transmissions simply drain more energy than needed, so their reception should be avoided.

Since reducing energy consumption is the critical issue in sensor networks, many authors have proposed protocols for energy-efficient broadcast: CDS (Connected Dominating Sets) [7], MPR (Multi Point Relays) [8], or RNG (Relative Neighborhood Graphs) [9]. They try to reduce energy consumption of the network by limiting the number of required transmissions or by performing power control to reduce transmission power. However, to the best of our knowledge, there is no protocol that attempts to reduce the number of redundant receptions. Although limiting the number of required transmissions already implies less redundant receptions, we propose to reduce redundant receptions even further to save more energy.

To reduce energy consumption during the reception of broadcast messages, we propose *abstract frames*: an abstract frame is a small control frame sent before each data frame. It contains a digest of the subsequent data frame. A node listening to the channel uses the information in the abstract frame to identify and filter out redundant messages at the MAC layer. If the node has already received the data frame, it can switch its radio off and save energy.

The rest of the paper is organized as follows. In Section 2, we present the key idea of abstract frames. In Section 3, we analyze the performance of our abstract frames method in terms of lifetime extension compared to two MAC protocols: an ideal one that totally avoids idle listening, but does not filter out redundant messages at the MAC layer and a practical one, SMAC [3]. In Section 4, we report simulation results on the performance of abstract frames method when used with SMAC.

2 Abstract Frames

An abstract frame is a small frame sent immediately before each broadcast frame. It contains a digest of the data in the subsequent broadcast frame. A node uses the information in the abstract frame to learn about the subsequent data contents. If a node learns from the abstract frame that the data frame has already been received, it can switch its radio off, because the subsequent data is redundant as shown in Fig. 1. In this way, a node only overhears redundant abstract frames instead of overhearing redundant data frames, which contributes to save more energy since abstract frames are expected to be far shorter than data frames.

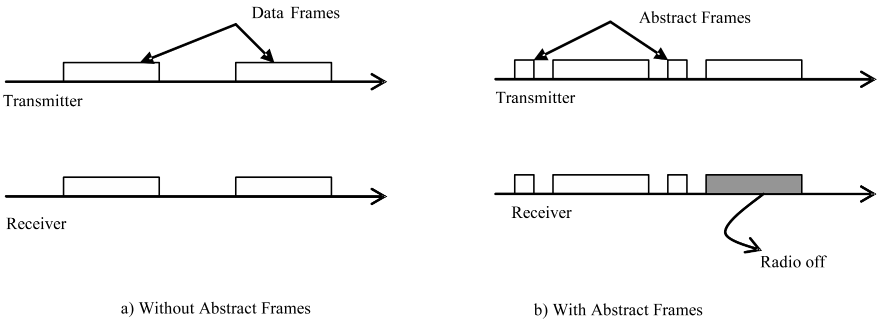


Fig. 1. Avoiding redundant frame reception by means of Abstract Frames

An abstract frame has a field that contains either a unique identifier, or a hash of the data contained in the subsequent broadcast frame. When the MAC layer needs to transmit a frame, it constructs and transmits the corresponding abstract frame before transmitting the broadcast frame. It inserts the hash field of the abstract frame in a table to avoid receiving it again from its neighbors. This table logs frames that have been recently seen so the MAC layer may switch the radio off when it expects a redundant reception.

According to this procedure, the MAC layer always receives an abstract frame before a data frame for broadcast communications. It first checks in its table whether there is an entry with the same hash value. If the entry exists, then the node switches its radio off to avoid receiving the same data again. If there is no such entry in the table, then the node continues to listen to the channel in order to receive the subsequent data frame. Once it has received the data frame, the node updates its table to avoid receiving redundant transmissions of the received data frame.

One can argue that there will be collisions in computing the hash value leading to false positives so that a node may ignore a data frame that has not been previously received. However, we think that this situation is unlikely for the following reasons. First, hash-field entries in the table of the MAC layer are not permanent, but cleaned after a timeout value. Second, we can choose a hash function and the size of the digest so that collisions are very rare. A frame will only be missed if it involves two simultaneously active colliding broadcasts during the timeout. Note that because broadcasts are not acknowledged, they are usually unreliable anyway.

There are several ways for reducing energy consumption caused by broadcasts. The most immediate one is to reduce the number of transmitted frames by avoiding redundant transmissions. Many proposed protocols select only a subset of nodes to flood a message while ensuring that all nodes eventually receive the message: CDS (Connected Dominating Sets) [7], MPR (Multi Point Relays) [8], or RNG (Relative Neighborhood Graphs) [9]. Another approach tries to optimize the transmission range by seeking a good trade-off between consuming more energy required to reach farther nodes or having more retransmissions [10, 11, 12]. We can apply the abstract frame approach to all such protocols, because it reduces energy consumption at the MAC layer. As all of them try to reduce broadcast traffic, the more efficient they are, the less abstract frames are necessary.

3 Theoretical Performance

Although the use of abstract frames results in less energy consumption during the reception of redundant frames, it also increases the energy drained per transmitted data frame. Therefore, we propose to analyze the performance of abstract frames taking into account these two parameters together. We propose to compare the lifetime of a node running a MAC protocol without abstract frames, which we call protocol P, to the lifetime of the same node when running protocol P with abstract frames, which we call P'.

In our analysis, we consider that all the communications are broadcast and nodes forward frames according to the flooding algorithm. As a candidate for protocol P, we take two examples. The first protocol is an ideal MAC protocol that does not have idle listening. The second protocol is SMAC that reduces idle listening via active/sleep schedules. For the sake of simplicity, we do not consider collisions in the following analysis.

We call E_P (resp. $E_{P'}$) the energy drained during a complete local flooding operation when nodes use protocol P (resp. P'). To quantify the ratio of lifetime extension by protocol P' compared to protocol P, we calculate gain G_P defined in (1) for the two candidate MAC protocols.

$$G_P = \frac{E_P}{E_{P'}} \tag{1}$$

3.1 Ideal MAC

To calculate the lifetime of a node, we consider the complete local flooding operation consisting of the reception of all frames from its neighbors and forwarding the broadcast frame exactly once. Therefore, if the node has n neighbors, then the energy drained during the flooding operation is:

$$E_{ideal} = nTP_r + TP_t, \tag{2}$$

where T is the transmission time of the data frame and P_t (resp. P_r) is the power drained by a transmission (resp. a reception).

When abstract frames are used, the node receives all abstract frames, but only one data frame. This is because the node discards the other data frames since they are redundant. In addition to that, the node transmits one abstract frame that precedes the data frame. The energy drained in this case is thus:

$$E_{ideal'} = (nA + T)P_r + (T + A)P_t, \tag{3}$$

where A is the transmission time of an abstract frame. Finally, the lifetime extension is the following:

$$G_{ideal} = \frac{E_{ideal}}{E_{ideal'}} = \frac{(n+\rho)T}{(1+\rho)T+(n+1)A}, \tag{4}$$

where $\rho = P_t/P_r$.

From Eq. (4), we conclude that the lifetime extension increases when data size increases and it converges to $\frac{n+\rho}{1+\rho}$ when $t \to \infty$. Also, when the number of neighbors increases, the gain increases up to $\frac{T}{A}$ when $n \to \infty$.

Note that the performance obtained with the use of abstract frames depends on the ratio $\frac{A}{T}$ of the abstract frame transmission time to the transmission time of data frames: the smaller A compared with T, the larger lifetime extension we get. We can calculate A_{max}, the maximum value of A beyond which there is no gain in using abstract frames: we need that $G_{ideal} > 1$, which leads to the following:

$$A_{max} = \left(\frac{n-1}{n+1}\right)T \tag{5}$$

3.2 SMAC

We follow the same methodology to evaluate the lifetime extension ratio for SMAC. Fig. 2 shows an example of a node with three neighbors. In SMAC, a node alternates active periods during which nodes can communicate and sleep periods during which nodes switch their radios off to save energy. The ratio of the period durations is controlled by the MAC duty-cycle parameter. The duration of the active period depends on a couple of parameters such that the data transmission time and the slot time used in the backoff procedure when nodes contend for the channel. Note that SMAC protocol chooses carefully the duration of the active period so that it is large enough to hold a data transmission including contention. However, there is no guarantee that the local flooding operation fits a single active period[1].

Let us call D the duration of the active period. The local flooding operation may fit one active period or more, depending on many parameters like the number of nodes n. Let us assume that a node needs k active periods, $(k > 0)$ to carry out the local flooding operation. Therefore, we have the following relations (see Fig. 2):

$$E_{smac} = nTP_r + TP_t + [kD - (n+1)T]P_i \tag{6}$$

and,

$$\begin{aligned} E_{smac'} = (T+nA)P_r + (T+A)P_t + (n-1)TP_s \\ +[kD - n(T+A) - (T+A)]P_i, \end{aligned} \tag{7}$$

where P_i (resp. P_s) is the power drained during the idle (resp. sleep) mode. In general, the power drained in the idle mode, in which the radio is ready to receive,

[1] For the sake of simplicity, Fig. 2 shows that the local flooding operation fits one active period.

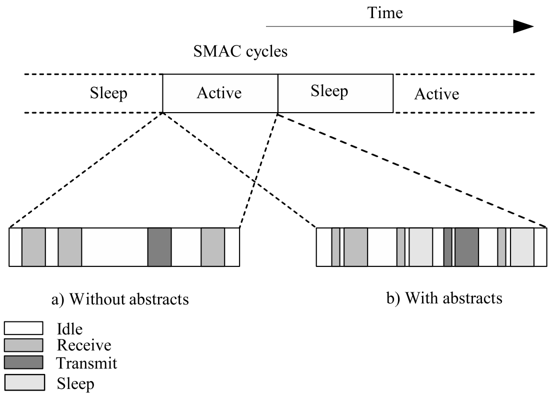


Fig. 2. Operation of SMAC with and without Abstract Frames

is slightly less than the power drained during the receive mode. However, to simplify the comparisons, we will assume that $P_i = P_r$. Also, the power drained during the sleep mode is negligible compared to other modes, so we assume that $P_s = 0$. By denoting $\rho = P_t/P_r$, we obtain the following lifetime extension for SMAC:

$$G_{smac} = \frac{E_{smac}}{E_{smac'}} = \frac{T\rho + kD - T}{(T + A)\rho + kD - nT - A} \tag{8}$$

Eq. (8) shows that the gain depends on kD, the duration of active periods needed for the complete local flooding. These active periods include idle listening. To show the effect of idle listening on lifetime extension, we propose to rewrite Eq. (8) in function of I, the amount of idle listening during the complete local flooding:

$$I = kD - nT - T \tag{9}$$

Thus, Eq. (8) can be rewritten as:

$$G_{smac} = \frac{(n + \rho)T + I}{(1 + \rho)T - A(1 - \rho) + I} \tag{10}$$

Eq. (10) shows that the gain decreases when idle listening increases. This is quite expected, because when idle periods dominate, then we will not get significant lifetime extension since the power drained during idle listening will be the same whether abstract frames are used or not.

Note that Eq. (10) implicitly includes the effect of traffic load on lifetime extension because idle listening depends on traffic load. Indeed, when traffic load is high, nodes spend more time in transmit and receive modes, which decrease the amount of idle listening. Therefore, we conclude that lifetime extension increases with traffic load. However, an excessive traffic load causes collisions that actually decreases the lifetime extension. We study this factor through simulation in Section 4.

Using the same approach as in Section 3.1, we calculate A_{max} for SMAC. We have,

$$A_{max} = \left(\frac{n-1}{\rho-1}\right)T \tag{11}$$

Interestingly, when $\rho \leq 1$, we have no constraint on the abstract frame transmission time to obtain a lifetime extension. Eq. (10) shows that the gain will always be larger than 1, provided $\rho \leq 1$. In this case, the power drained in the transmit mode is less than the power drained in the idle mode and transmitting abstract frames saves more energy than by staying idle.

4 Simulation

We have used ns-2 [13] to quantify the lifetime extension achieved with the use of abstract frames by simulation. We have chosen SMAC to represent a low power MAC protocol mainly because its code is public and seems to be stable. However, the idea of avoiding redundant frames reception by means of abstract frames may apply to a large variety of MAC protocols.

We compare the lifetimes achieved by two MAC protocols: SMAC without abstract frames and SMAC', which is SMAC with abstract frames. We carry out simulations to get more insight into the energy saving ratio since we have not taken all the parameters into account in the mathematical analysis.

Our application consists of a simple flooding agent that forwards a new broadcast message it receives exactly once (we assume that the size of data frames is fixed). The simulation scenario consists of one source node broadcasting messages according to a given traffic load. The other nodes flood the received broadcast message. The source node assigns a different message identifier for each new broadcast message. This identifier is used as our digest in the abstract frame. When the SMAC' agent receives an abstract frame, it checks whether it has already seen (sent or received) the data frame with the same message identifier. If the identifier is new, the SMAC' agent adds this identifier to an internal table and keeps the radio on to receive the subsequent data frame. However, if the identifier has been already seen, then the following data is redundant and the SMAC' agent switches the radio off to save energy.

The application agent counts the number of non-redundant received messages. We use this number to quantify the lifetime of a node. The ratio of the number of received messages with SMAC' out of the number of received messages with SMAC determines the lifetime extension. We have considered three situations to evaluate lifetime extension. These situations are the lifetime extension of the most vulnerable node, the lifetime extension of the most robust node, and the lifetime averaged over all the nodes. Results show that abstract frames extend the lifetime of SMAC with very close ratios in all theses three situations. Therefore, we have chosen to only analyze the results corresponding to the average number of messages in the rest of the discussion.

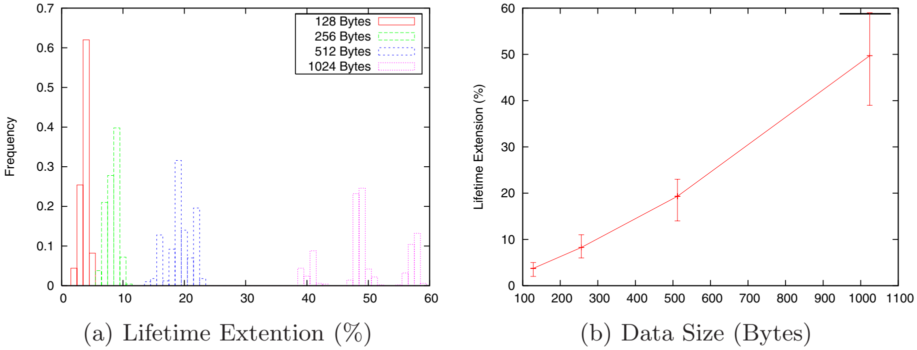
We measure the impact of data payload, traffic load, and network density on the lifetime extension. We use a simple energy model to simulate low power radio. In our energy model, the transmit mode uses 96mW, the receive mode uses 111mW and the ready-to-receive mode, that we also called idle mode, uses the same power as the receive mode.

4.1 Simple Star Network

For the topology, we take a simple star network with the source node placed at the center of the simulation area. The number of neighbors of the source node determines the density of the network.

For each simulation run, we calculate the lifetime extension as defined above. After some simulation runs, we calculate the average lifetime extension and the confidence interval corresponding to 95% of values. Note that the confidence interval is fairly large even with a very large set of simulation runs, which reflects the fluctuations in lifetime extension ratios due to the SMAC characteristics. In SMAC, the active period is composed of two periods: synchronization period and data period. Nodes use the synchronization period to exchange SYNC frames from time to time in order to synchronize on a common sleep/wakeup schedule. Note that SMAC does not guarantee that all nodes synchronize on a single common schedule. Nodes that share a common schedule form a virtual cluster and the network may contain one or several virtual clusters. Some nodes, called border nodes, may belong to more than one virtual cluster. There are many options to manage the active/wakeup schedules of the border nodes. In the implementation of SMAC we are using, a border node keeps active during all the active periods of the virtual clusters it belongs to. As a consequence, the actual MAC duty cycle of a border node increases, which decreases the efficiency of abstract frames since idle listening increases with the MAC duty cycle. Note that each time we run a simulation with a different random seed, we get different numbers of virtual clusters formed in the network. We have taken care to compare the lifetimes of SMAC and SMAC' in similar conditions, *i.e.*, for the same number of virtual clusters.

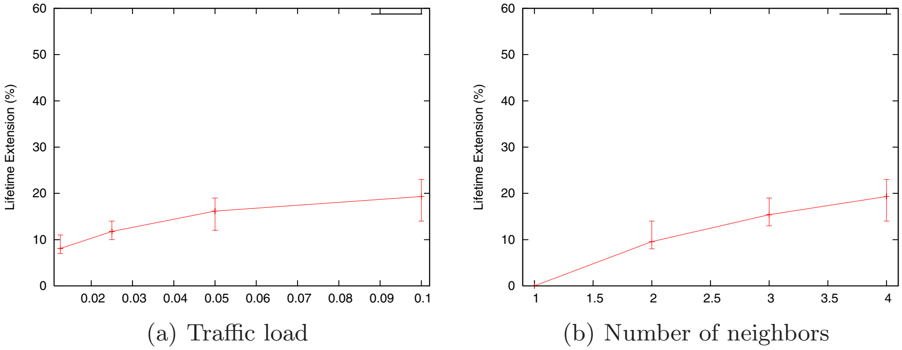
Fig. 3(a) shows the histogram of the lifetime extension for various data packet sizes. The variability is due to the number of virtual clusters formed during each



(a) Lifetime Extention (%)

(b) Data Size (Bytes)

Fig. 3. Lifetime extension with respect to the data size



(a) Traffic load (b) Number of neighbors

Fig. 4. Lifetime extension with respect to the traffic load and the number of neighbors

run. Fig. 3(b) presents a more legible view of the same data. Likewise, Fig. 4(a), 4(b), 5 show the lifetime extension as a function of other variables.

We notice that the lifetime extension is small for small data sizes, because the amount of time during which we switch the radio off to avoid redundant data reception becomes negligible (around 4%) compared to the time the radio is on. We have used the application duty-cycle that corresponds to the traffic load of 0.1 messages per second. However, when the data payload size increases to 1024 bytes, the lifetime extension increases by 40%-60%. This large interval is due to variations of the number of virtual clusters in each simulation run. As expected, the 40% ratio corresponds to the situation with many virtual clusters and the 60% ratio corresponds to fewer virtual clusters. Note that the formation of several virtual clusters decreases the lifetime extension because idle listening becomes significant. In [14], Li et al. pointed out this problem and proposed GSA (Global Schedule Algorithm) to make the network converge to one virtual cluster. We expect abstract frames to have better performance with GSA.

In Fig. 4(a), we have varied the traffic load from 0.0125 to 0.1 messages per second *i.e.*, from one message every 10 seconds to one message every 80 seconds. The payload of the messages is 512 bytes. The figure shows that the lifetime extension increases when the traffic load increases. In this case, the time during which data are exchanged becomes non negligible compared to the duration of idle listening. Hence, the duration during which SMAC' exploits abstract frames to switch off the radio becomes more significant. Note that we do not present results beyond 0.1 messages per second, because we have observed a considerable increase of collision rates for the traffic load larger than 0.1 and negligible energy saving ratios for the traffic load less than 0.0125. We argue that this is rather SMAC-dependent and not a result showing intrinsic low performance of abstract frames. We expect to get better performance with other low power MAC protocols that manage idle listening in a better way like TMAC [15], WiseMAC [16] and BMAC [17].

In Fig. 4(b), we have varied the number of neighbors of the source node to get different network degrees. This gives us a precise idea on what SMAC' is able

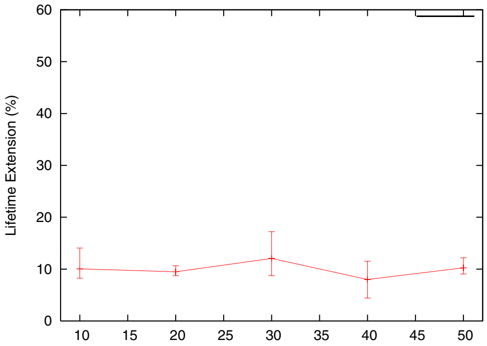


Fig. 5. Lifetime extension with respect to the number of nodes in the network

to achieve in situations in which the channel is not saturated and collisions are rare. For these simple star topology networks, we have observed a collision ratio less than 1%, which allows us to see the effect of network density on lifetime extension independently from collisions. As we expected, the lifetime extension increases with network density.

4.2 Realistic Network

In Fig. 5, we have measured the lifetime extension ratios for more realistic topologies. We have generated five networks with node positions distributed uniformly in a square area except for the source node always placed in the center. The networks are connected and the degree of a network with less nodes is less than the degree of a network with more nodes. The average densities of the networks are: 1.8, 2.7, 3.3, 3.6 and 4.12 for networks with 10, 20, 30, 40, and 50 nodes respectively. With the traffic load of 0.1 messages per second, we have observed collision rates increasing considerably.

Collisions mitigate the efficiency of abstract frames. Therefore, as opposed to what one may expect, the lifetime extension ratio may not systematically increase with the network density as shown in Fig. 5. This is because in most of the cases, collisions happen between abstract frames transmitted in the same time slot and receivers cannot correctly decode these abstract frames. Hence, it is not possible for receivers to know about the subsequent data contents so that they cannot switch their radios off. Note that this issue is rather related to the performance of the broadcast in 802.11-inspired MAC protocols. The way commonly used to decrease collision rates in these protocols is to increase the contention window size. However, this may be inefficient in SMAC, because increasing the contention window also increases idle listening, which is not desirable. Another reason of smaller gain ratios of abstract frames with SMAC is related to the formation of many virtual clusters. Indeed, when the density of nodes in network increases, border nodes will belong to more virtual clusters. Therefore, border nodes will be awaken during all the wakeup schedules of the virtual clusters they belong to, which increases idle listening and then mitigates the gain obtained with the use of abstract frames. For example, in networks with 10 nodes (resp. $20, ..., 50$) we have used in the simulation, each node belongs on the average to 1.3 (resp. 1.6, 1.8, 2.1 and 2.3) virtual clusters. We think that these mitigated gain ratios are rather due to SMAC and we expect larger gain ratios with protocols that manage idle listening better.

5 Conclusion

We have presented a novel idea for energy saving by reducing overhearing redundant copies of broadcast frames. It is based on abstract frames containing a digest of a subsequent data frame. We have evaluated of this scheme analytically and by means of simulation in ns-2. Although we have applied our approach to SMAC, the key idea is generic and can be used in a large variety of MAC protocols to further enhance energy savings of existing optimized broadcast protocols. We continue this work by evaluating the efficiency of abstract frames with other MAC protocols.

References

1. Ember Corporation. EM250 Single-Chip ZigBee/802.15.4 Solution, Data Sheet . 2005.
2. J. Polastre, R. Szewczyk, D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. *Proceedings of IPSN/SPOTS*, pages 302–11, Los Angeles, CA, April 2005.
3. W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. *Proceedings of the IEEE Infocom*, pages 1567–76, New York, NY, July 2002.
4. J. N. Al-Karaki, A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications* , 11(6):6–28, Dec 2004.
5. C. Intagonwiwat et al. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. on Networking*, 11(1):2–16, February 2003.
6. S.Y. Ni et al. The Broadcast Storm Problem in Mobile Ad Hoc Network. *Proceedings of the IEEE/ACM Mobicom*, 1999.
7. J. Wu and H. Li. On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks. *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 7–14, Aug 1999.
8. A. Qayyum, L. Viennot, A. Laouiti. Multipoint Relaying for Flooding Broadcast Messages in MobileWireless Networks. *Proceedings of IEEE HICSS*, Big Island, HI, Jan 2002.
9. G. Toussaint. The Relative Neighborhood Graph of a Finite Planar Set. *Pattern Recognition*, 12(4):261–8, 1980.
10. J. Cartigny, D. Simplot and I. Stojmenovic. Localized Minimum-Energy Broadcasting in Ad-hoc Networks. *Proceedings of the IEEE Infocom*, San Francisco, CA, April 2003.
11. J. Cartigny, et al. Localized LMST and RNG Based Minimum-energy Broadcast Protocols in Ad hoc Networks. *Ad Hoc Networks*, 3(1):1–16, 2004.

12. F. Ingelrest, D. Simplot-Ryl and I. Stojmenovic. Target Transmission Radius over LMST for Energy-Efficient Broadcast Protocol in Ad Hoc Networks. *Proceedings of IEEE ICC*, Paris, France, June 2004.
13. http://www.isi.edu/nsnam/ns/.
14. Y. Li, W. Ye, and J. Heidemann. Energy and Latency Control in Low Duty Cycle MAC Protocols. *Proceedings of the IEEE WCNC*, New Orleans, LA, March 2005.
15. T. van Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. *Proceedings of the ACM Sensys*, pages 171–80, Los Angeles, CA, November 2003.
16. Enz, C.C.; El-Hoiydi, A.; Decotignie, J.; Peiris, V. WiseNET: An Ultralow-Power Wireless Sensor Network Solution. *IEEE Computer*, 37(8):62–70, August 2004.
17. J. Polastre, J. Hill and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. *Proceedings of the ACM SenSys*, 2004.

Dynamic Resource Allocation in Communication Networks*

Antonio Capone[1], Jocelyne Elias[2], Fabio Martignon[3], and Guy Pujolle[2]

[1] Department of Electronics and Information, Politecnico di Milano
capone@elet.polimi.it
[2] University of Paris 6, LIP6 Laboratory,
8 rue du Capitaine Scott, 75015, Paris, France
jocelyne.elias@lip6.fr, guy.pujolle@lip6.fr
[3] Department of Management and Information Technology,
University of Bergamo, Italy
fabio.martignon@unibg.it

Abstract. Efficient dynamic resource provisioning algorithms are necessary to the development and automation of Quality of Service (QoS) networks. The main goal of these algorithms is to offer services that satisfy the QoS requirements of individual users while guaranteeing at the same time an efficient utilization of network resources. In this paper we introduce a new service model that provides quantitative per-flow bandwidth guarantees, where users subscribe for a guaranteed rate; moreover, the network periodically individuates unused bandwidth and proposes short-term contracts where extra-bandwidth is allocated and guaranteed exclusively to users who can exploit it to transmit at a rate higher than their subscribed rate. To implement this service model we propose a dynamic provisioning architecture for intra-domain Quality of Service networks. We develop an efficient bandwidth allocation algorithm that takes explicitly into account traffic statistics to increase the users' benefit and the network revenue simultaneously. We demonstrate through simulation in realistic network scenarios that the proposed dynamic provisioning model is superior to static provisioning in providing resource allocation both in terms of total accepted load and network revenue.

Keywords: Dynamic Bandwidth Allocation, Autonomic Networks, Service Model.

1 Introduction

Efficient dynamic resource provisioning mechanisms are necessary to the development and automation of Quality of Service networks. In telecommunication networks, resource allocation is performed mainly in a static way, on time scales

* This work is partially supported by the National Council for Scientific Research in Lebanon.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 892–903, 2006.

on the order of hours to months. However, statically provisioned network resources can become insufficient or considerably under-utilized if traffic statistics change significantly [1].

Therefore, a key challenge for the deployment of Quality of Service networks is the development of solutions that can dynamically track traffic statistics and allocate network resources efficiently, satisfying the QoS requirements of users while aiming at maximizing, at the same time, resource utilization and network revenue. Recently, dynamic bandwidth allocation has attracted research interest and many algorithms have been proposed in the literature [1, 2, 3, 4, 5]. These approaches and related works are discussed in Section 2.

Since dynamic provisioning algorithms are complementary to admission control algorithms [1], in our work we assume the existence of admission control algorithms at the edge of the network that cooperate with our proposed bandwidth allocation algorithm operating inside the network.

In this paper we propose a new service model that provides quantitative per-flow bandwidth guarantees, where users subscribe for a guaranteed transmission rate. Moreover, the network periodically individuates unused bandwidth and proposes short-term contracts where extra-bandwidth is allocated and guaranteed exclusively to users who can better exploit it to transmit at a rate higher than their subscribed rate.

To implement this service model we propose a distributed provisioning architecture composed by core and edge routers; core routers monitor bandwidth availability and periodically report this information to ingress routers using signalling messages like those defined in [2]. Moreover, if persistent congestion is detected, core routers notify immediately ingress routers.

Ingress routers perform a dynamic tracking of the effective number of active connections, as proposed in [6], as well as of their actual sending rate. Based on such information and that communicated by core routers, ingress routers allocate network resources dynamically and efficiently using a modified version of the max-min fair allocation algorithm proposed in [7]. Such allocation is performed taking into account users' profile and willingness to acquire extra-bandwidth based on their bandwidth utility function. The allocation is then enforced by traffic conditioners that perform traffic policing and shaping.

We evaluate by simulation the performance of our proposed bandwidth allocation algorithm in realistic network scenarios. Numeric results show that our architecture allows to achieve better performance than statically provisioned networks both in terms of accepted load and network revenue.

In summary, this paper makes the following contributions: the definition of a new service model and the proposition of a distributed architecture that performs dynamic bandwidth allocation to maximize users utility and network revenue.

The paper is structured as follows: Section 2 discusses related work; Section 3 presents our proposed service model and provisioning architecture; Section 4 describes the proposed dynamic bandwidth allocation algorithm; Section 5 discusses simulation results that show the efficiency of our dynamic resource allocation algorithm compared to a static allocation technique. Finally, Section 6 concludes this work.

2 Related Work

The problem of bandwidth allocation in telecommunication networks has been addressed in many recent works. In [7] a max-min fair allocation algorithm is proposed to allocate bandwidth equally among all connections bottlenecked at the same link. In our work we extend the max-min fair allocation algorithm proposed in [7] to perform a periodical allocation of unused bandwidth to users who expect more than their subscribed rate.

Dynamic bandwidth provisioning in Quality of Service networks has recently attracted a lot of research attention due to its potential to achieve efficient resource utilization while providing the required quality of service to network users [1, 2, 3, 4].

In [1], the authors propose a dynamic core provisioning architecture for differentiated services IP networks. The core provisioning architecture consists of a set of dynamic node and core provisioning algorithms for interior nodes and core networks, respectively. The node provisioning algorithm adopts a self-adaptive mechanism to adjust service weights of weighted fair queuing schedulers at core routers while the core provisioning algorithm reduces edge bandwidth immediately after receiving a Congestion-Alarm signal from a node provisioning module and provides periodic bandwidth re-alignment to establish a modified max-min bandwidth allocation to traffic aggregates.

The work discussed in [1] has similar objectives to our dynamic bandwidth allocation algorithm. However, their service model differs from our proposed model and traffic statistics are not taken into account in the allocation procedure. Moreover, in our work we suggest a distributed architecture implementation, while in these papers only a centralized scheme is considered.

A policy-based architecture is presented in [3], where a measurement-based approach is proposed for dynamic Quality of Service adaptation in DiffServ networks. The proposed architecture is composed of one Policy Decision Point (PDP), a set of Policy Enforcement Points that are installed in ingress routers and bandwidth monitors implemented in core routers. When monitors detect significant changes in available bandwidth they inform the PDP which changes dynamically the policies on in-profile and out-of-profile input traffics based on the current state of the network estimated using the information collected by the monitors. However, this scheme, while achieving dynamic QoS adaptation for multimedia applications, does not take into account the users utility function and their eventual willingness to be charged for transmitting out of profile traffic, thus increasing network revenue.

In [2], a generic pricing structure is presented to characterize the pricing schemes currently used in the Internet, and a dynamic, congestion-sensitive pricing algorithm is introduced to provide an incentive for multimedia applications to adapt their sending rates according to network conditions. As in [2], we take into account users bandwidth utility functions to evaluate our proposed allocation algorithm based on the increased network revenue that is achieved. However, the authors consider a different service model than that proposed in our work and focus mainly on the issue of dynamic pricing to perform rate adaptation based on network conditions.

The idea of measuring dynamically the effective number of active connections as well as their actual sending rate is a well accepted technique [4, 6]. In [4], the authors propose an active resource management approach (ARM) for differentiated services environment. The basic concept behind ARM is that by effectively knowing when a client is sending packets and how much of its allocated bandwidth is being used at any given time, the unused bandwidth can be reallocated without loss of service. This concept is in line with our proposed bandwidth allocation algorithm. Differently from our work, however, ARM does not guarantee to the user a minimum subscribed bandwidth throughout the contract duration since unused bandwidth is sent to a pool of available bandwidth and it can be used to admit new connections in the network, in spite of those already admitted.

3 Service Model and Dynamic Provisioning Architecture

We first introduce our proposed service model, then we present a distributed provisioning architecture which implements such service model by performing the dynamic bandwidth allocation algorithm described in Section 4; finally, we present the signalling messages used to assure the interaction between network elements.

3.1 Service Model

We propose a service model that, first, provides a quantitative bandwidth guarantee to users and then exploits the unused bandwidth individuated periodically in the network to propose short-term guaranteed extra-bandwidth. In this process, different weights can be assigned to network users to allocate extra-bandwidth with different priorities; such weights can be set statically offline, based on the service contract proposed to the user, or can be adapted on-line based, for example, on the user bandwidth utility function.

Our proposed service model is therefore characterized by:

- a quantitative bandwidth guarantee, expressed through the specification of user's subscribed rate;
- short term guaranteed extra-bandwidth: the network is monitored on-line to individuate unused bandwidth that is allocated with guarantee, during the update interval, to users who can exploit it to transmit extra-traffic;
- a weight that expresses the user's priority in the assignment of extra-bandwidth;
- a bandwidth utility function, $U(x)$, that describes the user's preference for an allocation of x bandwidth units. In line with [8] we consider the utility function as part of the service model. Without loss of generality, we do not consider the pricing component of a bandwidth utility function.

3.2 Architecture and Control Messaging

To implement our service model we assume a distributed architecture constituted by core and edge routers, as shown in Fig.1; traffic monitors are installed on ingress and core routers to perform on-line measurements on the incoming traffic flows and network capacity utilization, respectively.

Core routers exchange messages with ingress routers to report the link utilization or to notify a congestion situation. Each ingress router collects the measurements performed by traffic monitors and exchanges periodically update messages with all other ingress routers to report the current incoming traffic statistics. Moreover, a dynamic bandwidth allocation algorithm is implemented in all ingress routers: it takes into account the traffic statistics gathered at ingress routers and the network information reported by core routers to allocate network resources dynamically and efficiently.

The messages exchanged between network routers, illustrated with arrows in Fig.1, are similar to the control messages that have been proposed in [1] to report persistent congestion or resource availability.

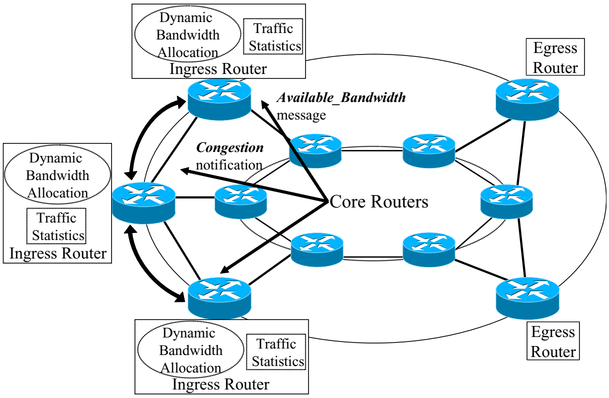


Fig. 1. The proposed distributed architecture that supports dynamic bandwidth allocation

4 Dynamic Bandwidth Allocation Algorithm

We propose a novel dynamic provisioning algorithm that allocates network capacity efficiently based on traffic statistics measured on-line. Bandwidth allocation is performed by ingress routers periodically and is enforced using traffic conditioners. We denote the interval between two successive allocations performed by the algorithm as the *update interval*, whose duration is T_u seconds. Moreover, core routers monitor link utilization, and if congestion on some links is detected, bandwidth re-allocation is immediately invoked to solve this situation.

In the following we present in details the bandwidth allocation algorithm, that proceeds in two steps: in the first step, bandwidth is allocated to all active connections trying to match their near-term traffic requirements that are predicted based on statistics collected by ingress routers. In step two, spare bandwidth as well as bandwidth left unused by idle and active connections is individuated on each link. Such available extra-bandwidth is allocated with guarantee during the current update interval exclusively to connections that can take advantage of it since they are already fully exploiting their subscribed rate.

To illustrate the allocation algorithm, let us model the network as a directed graph $G = (N, L)$ where nodes represent routers and directed arcs represent links. Each link $l \in L$ has associated the capacity C_l. A set of K connections is offered to the network. Each connection is represented by the notation (s_k, d_k, sr_k), for $k = 1, \ldots, K$, where s_k, d_k and sr_k represent the connections source node, destination node and the subscribed rate, respectively; furthermore, we assume that each connection has associated r_min_k, which represents the minimum bandwidth the application requires. Let a_k^l be the routing matrix: $a_k^l = 1$ if connection k is routed on link l, $a_k^l = 0$ otherwise. We assume that a communication between a user pair is established by creating a session involving a path that remains fixed throughout the user pair conversation duration. The session path choice method (i.e., the routing algorithm) is not considered in this paper.

At the beginning of the $n - th$ update interval, each ingress router computes the transmission rate, b_k^{n-1}, averaged over the last T_u seconds, for all connections $k \in K$ that access the network through it. This information is then sent to all other ingress routers using control messages as described in the previous Section, so that all ingress routers can share the same information about current traffic statistics and perform simultaneously the same allocation procedure.

The amount of bandwidth allocated to each source k during the $n - th$ update interval, r_k^n, is determined using the two-steps approach described in the following:

- First step: Connections having $b_k^{n-1} < r\_min_k$ are considered *idle*; all other active connections are further classified as *greedy* if they used a fraction greater than γ of their subscribed rate sr_k (i.e. if $b_k^{n-1} > \gamma \cdot sr_k$), otherwise they are classified as $non - greedy$. In our implementation we set $\gamma = 0.9$.

  Let us denote by K_i, K_{ng} and K_g the sets of idle, non-greedy and greedy connections, respectively.

  Idle connections are assigned their minimum required transmission rate, i.e. $r_k^n = r\_min_k, \forall k \in K_i$.

  Non-greedy connections are assigned a bandwidth that can accommodate traffic growth in the current update interval while, at the same time, save unused bandwidth that can be re-allocated to other users. Several techniques have been proposed in the literature to predict the near-term transmission rate of a connection based on past traffic measurements. In this work we only consider the last measured value, b_k^{n-1}, and we propose the following simple bandwidth allocation: $r_k^n = \min\{2 \cdot b_k^{n-1}, sr_k\}, \forall k \in K_{ng}$. In this regard we are currently studying more efficient traffic predictors that could allow improved bandwidth allocation.

Greedy connections are assigned in this step their subscribed rate sr_k, and they also take part to the allocation of extra-bandwidth performed in step two, since they are already exploiting all their subscribed rate.

- Second step: after having performed the allocations described in step one, the algorithm individuates on each link l the residual bandwidth R_l, i.e. the spare bandwidth as well as the bandwidth left unused by idle and non-greedy connections. R_l is hence given by the following expression:

$$R_l = C_l - (\sum_{k \in K_i \cup K_{ng}} r_k^n \cdot a_k^l + \sum_{k \in K_g} sr_k \cdot a_k^l), \forall l \in L \tag{1}$$

where the first summation represents the total bandwidth allocated in step one to idle and non-greedy connections, while the second summation represents the bandwidth allocated to greedy connections.

Such extra-bandwidth is distributed exclusively to greedy connections using the algorithm detailed in Table 1, which is an extension of the allocation algorithm proposed in [7]. This algorithm takes as input the set K_g of greedy connections, the link set L with the residual capacity on each link l, R_l, and the routing matrix a_k^l, and produces as output the amount of extra-bandwidth $f_k^n, k \in K_g$ that is assigned to each greedy connection during the $n - th$ update interval, so that finally $r_k^n = sr_k + f_k^n, \forall k \in K_g$.

To take into account users weights it is sufficient to substitute n_l in Table 1 with w_l, which is defined as the sum of the weights of all greedy connections that are routed on link l.

Table 1. Pseudo-code specification of the bandwidth allocation algorithm

```
(1) initiate all $f_k^n = 0, \ \forall \ k \in K_g$
(2) remove from the link set $L$ all links $l \in L$ that have
    a number of connections crossing them $n_l$ equal to 0
(3) for every link $l \in L$, calculate $F_l = R_l/n_l$
(4) identify the link $\alpha$ that minimizes $F_\alpha$
    i.e. $\alpha \mid F_\alpha = min_k(F_k)$
(5) set $f_k^n = F_\alpha, \ \forall \ k \in K_\alpha$, where $K_\alpha \subseteq K_g$ is the set
    of greedy connections that cross link $\alpha$
(6) for every link $l$, update the residual capacity and the
    number of crossing greedy connections as follows:

        $R_l = R_l - \sum_{k \in K_\alpha} f_k^n \cdot a_k^l$
        $n_l = n_l - \sum_{k \in K_\alpha} a_k^l$

(7) remove from set $L$ link $\alpha$ and those that have $n_l = 0$
(8) if $L$ is empty, then stop; else go to Step (3)
```

It should be clarified that our algorithm can temporarily present some limitations in bandwidth allocation, since the bandwidth allocated to a user can at most double from an update interval to the successive one. This could affect the performance of users that experience steep increases in their transmission rate. In Section 5 we evaluate numerically this effect showing at the same time how it is counterbalanced by increased network revenue in all the considered network scenarios under several traffic load conditions.

5 Numeric Results

In this Section we compare the performance, measured by the average accepted load and network extra-revenue versus the total load offered to the network, of the proposed dynamic bandwidth allocation algorithm with a static provisioning strategy, referring to different network scenarios to cover a wide range of possible environments. Static provisioning allocates to each source k its subscribed rate sr_k.

We are interested in measuring the following performance metrics: the average accepted load and network extra-revenue. The average accepted load is obtained averaging the total load accepted in the network over all the bandwidth update intervals.

We define, in line with [2], the average network extra-revenue as the total charge paid to the network for all the extra-bandwidth utilization, averaged over all the bandwidth update intervals. In this computation we consider only network extra-revenue generated by greedy users that are assigned extra-bandwidth by our proposed dynamic allocation algorithm. Furthermore we assume, in line with [5], that the utilities are additive so that the aggregate utility of rate allocation is given by the sum of the utilities perceived by all network users.

Using the notation introduced in the previous section, the average network extra-revenue can be obtained averaging over all the update intervals n the quantity:

$$\sum_{k \in K_g} U(b_k^n) - U(sr_k) \tag{2}$$

In the first scenario we gauge the effectiveness of the proposed traffic-based bandwidth allocation algorithm. We consider, in line with [1, 2], the scenario illustrated in Figure 2, that consists of a single-bottleneck with 12 source-destination pairs. All links are full-duplex and have a propagation delay of 1 ms. The capacities of the links are given in the figure.

We use 12 Exponential On-Off traffic sources; the average On time is set to 200 s, and the average Off time is varied in the 0 to 150 s range to simulate different traffic load conditions while at the same time varying the percentage of bandwidth left unused by every connection. Six sources have a peak rate of 40 kb/s and a subscribed rate of 100 kb/s while the remaining sources have a peak rate of 1 Mb/s and a subscribed rate of 300 kb/s; the minimum bandwidth required by each source, r_min_k, is equal to 10 kb/s. The algorithm updating interval, T_u, is set equal to 20 s. We assume, for simplicity, that all users have the

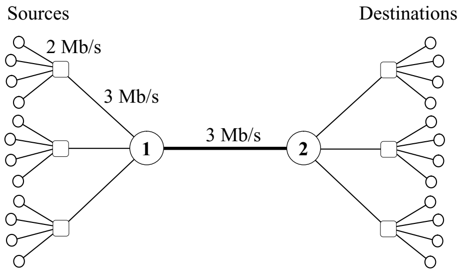


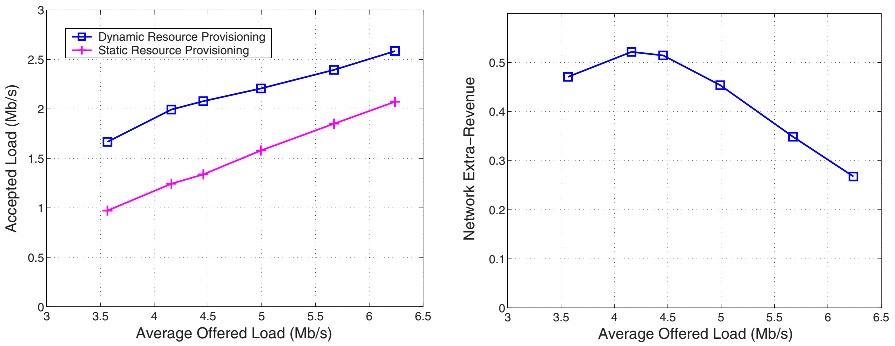
Fig. 2. Network topology with a single bottleneck

same weight w_k and the same utility function proposed in [8], $U(x) = 1 - e^{\frac{x^2}{x+h}}$, that models the perceived utility of real-time elastic traffic for an allocation of x bandwidth units. The parameter h setting is the same as in [8].

Note that a realistic characterization of network applications is outside the scope of this paper. The specification of the utility function allows us exclusively to gauge the extra network revenue that can derive from the deployment of our proposed bandwidth allocation algorithm.

Figures 3(a) and 3(b) show, respectively, the average total load accepted in the network and the corresponding total extra-revenue as a function of the average total load offered to the network. It can be observed that our dynamic provisioning algorithm is very efficient in resource allocation compared to a static provisioning algorithm for all values of the offered load, providing improvements up to 60% in the total accepted traffic.

The maximum network extra-revenue is achieved when the average Off time of exponential sources is equal to 100 s, corresponding to an offered load approximately equal to 4 Mb/s. In this situation, in fact, the average number of idle connections (i.e. 4) is sufficiently high to exalt our dynamic allocation algorithm



(a) (b)

Fig. 3. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the network of Fig. 2

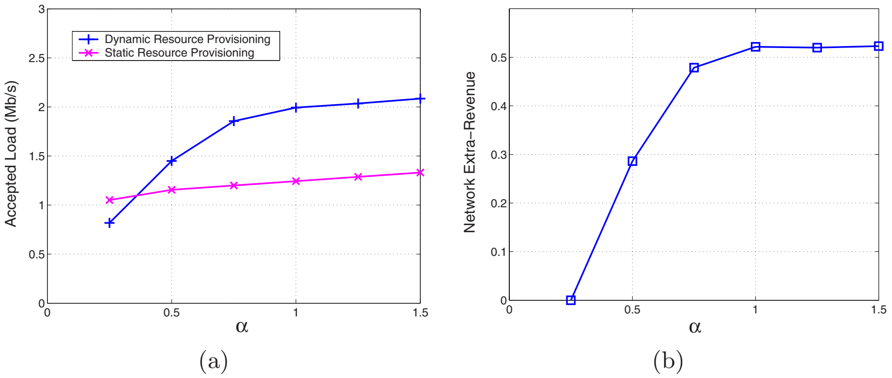


Fig. 4. Average total accepted load (a) and network extra-revenue (b) versus the rate scaling factor α in the network of Fig. 2

that reallocates unused bandwidth to active users who can take advantage of it, sending extra-traffic and generating network revenue. With lower Off time values (i.e. with higher offered loads) the total extra-revenue slightly decreases as less connections are idle, in average, and consequently less bandwidth is available for re-allocation.

To investigate the impact on the performance of the update interval duration, we have considered, in the same scenario, different values for T_u, i.e. 40 s and 60 s. We found that the average increase in the total accepted load, expressed as a percentage of the traffic admitted in the static allocation case, was of 32% for $T_u = 40$ s and 21% for $T_u = 60$ s, while for $T_u = 20$ s it was 47% (see Fig. 3(a)). These results allow to gauge the trade-off between performance improvement and overhead resulting from a more frequent execution of the allocation algorithm.

In the same scenario of Figure 2 we then fixed the average Off time of Exponential sources to 100 s while maintaining the average On time equal to 200 s, and we varied the peak rate of all sources scaling them by a factor α, with

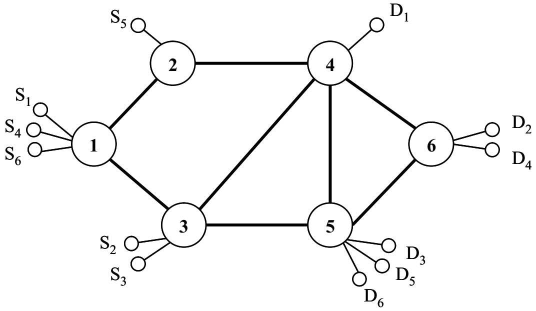


Fig. 5. Network topology with a larger number of links

Table 2. Peak rate, subscribed rate and path for the connections in the network scenario of Figure 5

Connection	Peak Rate (kb/s)	Subscribed Rate (kb/s)	Path
1	100	250	1-3-4
2	100	250	3-4-6
3	100	250	3-4-5
4	1000	500	1-2-4-6
5	1000	500	2-4-5
6	1000	1000	1-3-5

$0.25 \leq \alpha \leq 1.5$. Figures 4(a) and 4(b) show the total accepted load and the total extra-revenue in this scenario.

At very low load the static provisioning technique achieves slightly higher performance than dynamic provisioning. This is due to the fact that in this situation static provisioning is in effect sufficient to accommodate all incoming traffic; on the other hand, dynamic provisioning needs some time (in the worst case up to T_u seconds) to track the transition of sources from the idle to the active state. For all other traffic loads the advantage of the proposed dynamic bandwidth allocation algorithm is evident both in terms of accepted load and network extra-revenue.

A more realistic scenario is shown in Fig. 5. It comprises 6 nodes and 8 bidirectional links, all having a capacity equal to 2 Mb/s and propagation delay of 1 ms. In this topology, 6 Exponential On-Off traffic sources are considered, and their source and destination nodes are indicated in the Figure. Table 2 reports for all the connections the peak rate, the subscribed rate and the path of the connection. All other parameters are set as in the previous scenarios. Note that, with such paths choice, various connections compete for network capacity with different connections on different links.

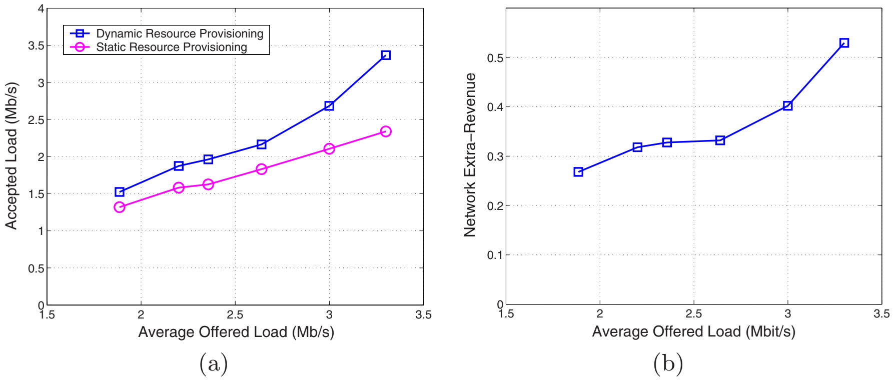


Fig. 6. Average total accepted load (a) and network extra-revenue (b) versus the average load offered to the network of Fig. 5

Also in this scenario the dynamic allocation algorithm outperforms static allocation, as shown in Figures 6(a) and 6(b), thus proving the benefit of the proposed scheme. These results verify that our allocation algorithm allows service providers to increase network capacity utilization and consequently network extra-revenue with respect to static provisioning techniques.

6 Conclusion

In this paper we proposed a novel service model where users subscribe for guaranteed transmission rates, and the network periodically individuates unused bandwidth that is re-allocated and guaranteed with short-term contracts to users who can better exploit it. We described a distributed dynamic resource provisioning architecture for quality of service networks. We developed an efficient bandwidth allocation algorithm that takes explicitly into account traffic statistics to increase the users perceived utility and the network extra-revenue.

Simulations results measured in realistic network scenarios show that our allocation algorithm allows to increase both resource utilization and network revenue with respect to static provisioning techniques.

References

1. A. T. Campbell and R. R.-F. Liao. Dynamic Core Provisioning for Quantitative Differentiated Services. *IEEE/ACM Transactions on Networking*, pages 429–442, vol. 12, no. 3, June 2004.
2. H. Schulzrinne and X. Wang. Incentive-Compatible Adaptation of Internet Real-Time Multimedia. *IEEE Journal on Selected Areas in Communications*, pages 417–436, vol. 23, no. 2, February 2005.
3. T. Ahmed, R. Boutaba, and A. Mehaoua. A Measurement-Based Approach for Dynamic QoS Adaptation in DiffServ Network. *Journal of Computer Communications, Special issue on End-to-End Quality of Service Differentiation, Elsevier Science*, 2004.
4. M. Mahajan, M. Parasharand, and A. Ramanathan. Active Resource Management for the Differentiated Services Environment. *International Journal of Network Management*, pages 149–165, vol. 14, no. 3, May 2004.
5. F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, pages 33–37, vol. 8, 1997.
6. J. Aweya, M. Ouellette, and D. Y. Montuno. Design and stability analysis of a rate control algorithm using the Routh-Hurwitz stability criterion. *IEEE/ACM Transactions on Networking*, pages 719–732, vol. 12, no. 4, August 2004.
7. D. Bertsekas and R. Gallager. *Data Networks, 2nd Edition.* Prentice-Hall, 1992.
8. L. Breslau and S. Shenker. Best-Effort versus Reservations: A Simple Comparative Analysis. In *Proc. ACM SIGCOMM*, pages 3–16, September 1998.

Fair Assured Services Without Any Special Support at the Core[⋆]

Sergio Herrería-Alonso, Manuel Fernández-Veiga, Andrés Suárez-González,
Miguel Rodríguez-Pérez, and Cándido López-García

Departamento de Enxeñería Telemática, Universidade de Vigo,
Campus universitario, 36310 Vigo, Spain
sha@det.uvigo.es

Abstract. Many users require IP networks with the capacity to guarantee a minimum throughput even during periods of congestion. Furthermore, it is also desirable to share the excess unsubscribed bandwidth among active users if aggregate demand does not exceed network capacity. This kind of service, named assured service, can be provided through the *Assured Forwarding* (AF) *Per Hop Behavior* (PHB) defined in the DiffServ architecture. DiffServ mechanisms require special networking support at both the edge and the core nodes to guarantee the differentiated service. In this paper we propose the *Ping Trunking* scheme as a suitable mechanism to provide assured services to network users without the need for modifying core nodes. *Ping Trunking* is an edge-to-edge management technique that completely addresses the regulation of aggregate traffic streams at the edge of the network. In addition, it also overcomes some unfairness issues found in AF when sharing the available bandwidth among heterogeneous aggregates. Simulation results have validated the effectiveness of our proposal.

Keywords: DiffServ, assured services, aggregated traffic, congestion control.

1 Introduction

The Internet best effort model with no service guarantee is no longer acceptable in view of the proliferation of interactive applications such as Internet telephony, video conferencing or networked games. The growing importance of these recent applications with stringent constraints behooves the research community to develop a new range of network services able to accommodate heterogeneous application requirements and user expectations.

Among the new services demanded, assured services are one of the most popular. Assured services must provide different levels of forwarding assurances for

[⋆] This work was supported by the "Ministerio de Educación y Ciencia" through the project TIC2003-09042-C03-03 of the "Plan Nacional de I+D+I" (partially financed with FEDER funds) and by the "Secretaría Xeral de Investigación de la Xunta de Galicia" through the grant PGIDT04PXIC32203PN.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 904–915, 2006.

IP packets. For instance, many users just require a guarantee that IP packets are forwarded with high probability as long as their aggregate traffic streams do not exceed their committed information rate. In addition, it is also desirable that users may exceed their subscribed profiles with the understanding that the excess traffic is not forwarded with as high probability as the traffic that is within the profile. This kind of services can be provided through the *Assured Forwarding* (AF) [1] *Per Hop Behavior* (PHB) defined in the DiffServ architecture [2]. The differentiated service is obtained through traffic conditioning and packet marking at the edge of the network along with differentiated forwarding mechanisms at the core. Consequently, every node in a DiffServ network must be adapted to provide the required differentiation.

In [3] we proposed a new edge-to-edge management scheme named *Ping Trunking* able to provide some service guarantees to aggregate traffic streams. Our proposal establishes a Vegas-like control connection between the ingress and egress node of each aggregate. This connection regulates the flow of the aggregate traffic stream into the core of the network enforcing congestion control for the managed aggregate at its ingress node. In this paper, we argue that *Ping Trunking* can be used to offer assured services without requiring any special support at the core. Our proposal addresses all control tasks at the edges of the network so that the core nodes do not need to support any particular function for service differentiation. This feature makes our proposal easily deployable and improves its interest considerably.

In addition, *Ping Trunking* also overcomes some unfairness issues found in AF when sharing the available bandwidth. Several studies have shown that the number of flows in aggregates, the round trip time, the mean packet size and the TCP/UDP interaction are key factors in the throughput obtained by competing aggregates [4, 5]. With the help of some simulation experiments, we show how our proposal is able to distribute the available bandwidth among heterogeneous aggregates in a fair manner.

The rest of the paper is organized as follows. Section 2 gives a brief overview of AF PHB. In Sect. 3, we illustrate the *Ping Trunking* mechanism. Section 4 describes the simulation configuration used for the evaluation of both techniques. In Sect. 5, we present the results obtained from the simulation experiments. Section 6 briefly describes other approaches proposed to improve fairness requirements for assured services. We end this paper with some concluding remarks and future lines in Sect. 7.

2 Assured Forwarding PHB

AF distinguishes four classes of delivery for IP packets and three levels of drop precedence per class. Each AF class has a certain amount of buffer space and bandwidth reserved in each node. Within each class, IP packets are marked based on conformance to their target throughputs. The *Time Sliding Window Three Color Marker* (TSWTCM) is one of the most interesting packet marking algorithms proposed to work with AF [6]. In this algorithm, two target rates are defined: the *Committed Information Rate* (CIR) and the *Peak Information Rate* (PIR). Under TSWTCM, the aggregated traffic is monitored and when the measured traffic is below its CIR, packets are marked with the lowest drop precedence, *AFx1*. If the measured traffic exceeds its CIR but falls below its PIR, packets are marked with a higher drop precedence, *AFx2*. Finally, when traffic exceeds its PIR, packets are marked with the highest drop precedence, *AFx3*.

At the core of the network, the different drop probabilities can be achieved using the RIO (*RED with In/Out*) scheme [7], an active queue management technique that extends RED gateways [8] to provide service differentiation. RIO is configured with three different sets of RED parameters, one for each of the drop precedence markings. These different RED parameters cause packets marked with a higher drop precedence to be discarded more frequently during periods of congestion than packets marked with a lower drop precedence.

3 Ping Trunking

Ping Trunking [3] is an edge-to-edge management technique that provides some service guarantees to aggregate traffic streams. Our proposal is based on a technique named *TCP Trunking* [9, 10], but we have extended and improved the original scheme to manage aggregates in a simpler and smoother way.

A ping trunk is an aggregate traffic stream where data packets are transmitted at a rate dynamically determined by a preventive congestion control algorithm. Each trunk carries a varying number of user flows for common treatment between two nodes of the network (the ingress and the egress nodes). The flow of the aggregated traffic is regulated by a single control connection established between the two edges of the trunk. This control connection injects control packets into the network to probe its congestion level. The introduction of control packets is not conditioned by the user data protocols, but it is only determined by the control connection itself. In addition, a trunk will not retransmit user packets if they are lost. If it is required, retransmissions should be handled by user applications on the end hosts.

Figure 1 provides greater detail on the operation of this mechanism. Incoming user packets at the ingress node are classified as belonging to a particular trunk and queued in the corresponding trunk buffer. User packets can only be forwarded when credit for their trunk is available. The credit value represents the amount of user data allowed to be forwarded. When a user packet is sent, the credit is decremented by the size of the packet. When a control packet is sent, the credit is incremented by the size of the control congestion window (*cwnd*). Therefore, the transmission of user data is regulated by both the forwarding of control packets and the *cwnd* value.

It is important to point out that both user and control packets must follow the same path between the edges of the trunk to ensure that control connections are probing the proper available bandwidth. This assumption can be absolutely guaranteed if trunks are run on top of ATM virtual circuits or MPLS label-switched paths [11].

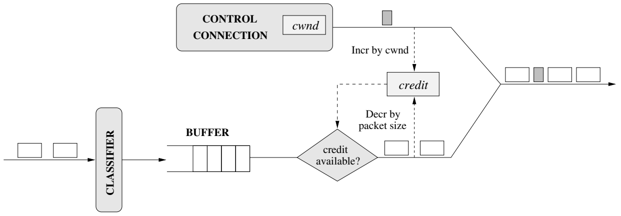


Fig. 1. Block diagram. This figure includes several simplifications. In fact, each trunk has its own buffer, credit bucket and control connection.

3.1 Operations of the Control Connection

The control connection is in charge of measuring the round-trip time (RTT) between the edges of the trunk accurately. Then, a Vegas-like congestion control mechanism will employ this RTT estimate when adapting the transmission rate of the trunk. Let us start by giving a brief description of the operations accomplished by this connection.

When the first user packet arrives to the ingress node of the trunk, a control packet is generated and sent. For each control packet that reaches the egress node of the trunk, its corresponding acknowledgment (ack) is generated. The arrival of an ack back at the ingress node triggers the transmission of a new control packet. Therefore, the control connection only sends control packets on reception of acks, i.e., once per RTT.

To avoid the starvation of control connections, a waiting time-out timer is needed. This timer is started every time a control packet is sent. If the ack of the last control packet sent does not arrive to the ingress node before the timer expires, the connection will consider that the packet has been lost and it will send a new control packet.[1]

Control connections use a method similar to that used in the TCP estimation of RTT. To carry out this task, they timestamp with its local time every control packet and this timestamp is echoed in the acks. The value of the last RTT sample observed is computed as the difference between the current time and the timestamp field in the ack packet. The RTT is eventually estimated using an exponential moving average taken over RTT samples.

3.2 Vegas-Like Congestion Control Mechanism

The transmission rate of each trunk should be able to update dynamically according to current network conditions. We propose the use of a Vegas-like congestion control mechanism to discover the available bandwidth that each trunk should obtain. TCP Vegas [12] is an implementation of TCP that employs proactive techniques to increase throughput and decrease packet losses. The congestion

[1] The control connections employed behave like the *ping* command used to send ICMP ECHO_REQUEST/REPLY packets to network hosts. Hence the name of our proposal.

control mechanism introduced by Vegas gives TCP the ability to detect incipient congestion before losses are likely to occur, so we have devised a similar mechanism adapted to trunks.

Upon receiving each ack, control connections calculate the expected throughput and the current actual throughput as in TCP Vegas. If it is assumed that trunks are not overflowing the path, the expected throughput can be calculated as $cwnd/d$, where d is the round-trip propagation delay and can be estimated as the minimum of all measured RTTs. On the other hand, the actual throughput is given by $cwnd/D$, where D is the RTT estimation. These throughputs are compared and then, control connections adjust their congestion windows accordingly. Let *Diff* be the difference between the expected and the actual throughput:

$$Diff = Expected - Actual = \left(\frac{cwnd}{d} - \frac{cwnd}{D}\right)d \ . \tag{1}$$

The *Diff* value has been scaled with the minimum RTT so that *Diff* can be seen as the amount of user data in transit.

There are two thresholds defined: α, β, with $\alpha \le \beta$. When $Diff < \alpha$, the trunk is allowed to increment its amount of user data in transit, and therefore, the control connection can increase its congestion window linearly. If $Diff > \beta$, the trunk is forced to decrease its congestion window linearly. In any other case, the congestion window remains unchanged.

This mechanism stabilizes the value of the congestion window and reduces packet drops. If a control packet loss is detected, the available bandwidth is halved, but this should happen sporadically.

3.3 Setting of Vegas Parameters

We should determine the suitable values of Vegas parameters that must be assigned to each trunk so that they can obtain their fair share of the available bandwidth. Consider a bottleneck shared by a set of trunks indexed by i. Let C denote the bottleneck capacity. Each trunk i has associated a subscribed target rate r_i. The overall demand R is the sum of the subscribed target rates for all active trunks. If $R < C$, the excess unsubscribed bandwidth should be distributed among trunks in proportion to the contracted target rates. Therefore, the fair bandwidth f that should be ideally allocated to each trunk i is obtained as

$$f_i = r_i + (C - R)\frac{r_i}{R} = \frac{r_i C}{R} \ , \tag{2}$$

where $R = \sum_i r_i$.

According to one interpretation of Vegas [13], congestion windows of control connections must satisfy the following equation in the equilibrium (we assume for simplicity that $\alpha_i = \beta_i$):

$$\left(\frac{cwnd_i}{d_i} - \frac{cwnd_i}{D_i}\right)d_i = \alpha_i \ . \tag{3}$$

On the other hand, the transmission rate x of a given trunk is determined by the *cwnd* value of its corresponding control connection ($cwnd = xD$). Substituting this in (3), we have

$$\left(\frac{x_i D_i}{d_i} - \frac{x_i D_i}{D_i}\right) d_i = \alpha_i \ , \tag{4}$$

and, from (4), it follows that

$$x_i = \frac{\alpha_i}{D_i - d_i} \ . \tag{5}$$

The RTT can be calculated as the sum of two delays: the round-trip propagation delay (d) and the queueing delay (B/C), where B denotes the total backlog buffered in the network. Then, from (5), and using $D_i = d_i + B/C$, the transmission rate of each trunk can be expressed as

$$x_i = \frac{\alpha_i C}{B} \ . \tag{6}$$

Finally, equating (2) and (6) yields the suitable value of α threshold that permits to allocate to each trunk its desired share of bandwidth:

$$\alpha_i = \frac{r_i B}{R} \ . \tag{7}$$

Therefore, to compute the α threshold, each trunk must know both B and R parameters. The B parameter should have a fixed low value set by the network manager to encounter small queues at the core. However, the necessity of determining the overall aggregated demand in all edge nodes may complicate our proposal substantially.

Fortunately, we can demonstrate that it is not required to know the value of the overall demand very accurately. Assume $R' \neq R$, $R' > 0$, was used as the aggregated demand. The total backlog B' actually buffered in the network is obtained as the sum of the α thresholds of all competing trunks. Then,

$$B' = \sum_i \alpha_i = \sum_i \frac{r_i B}{R'} = \frac{B}{R'} \sum_i r_i = \frac{BR}{R'} \ . \tag{8}$$

From (6), and using $B'R' = BR$ derived from (8), we can conclude that the fairness condition is still satisfied although the R' value employed is false:

$$x_i = \frac{\alpha_i C}{B'} = \frac{r_i BC}{R'B'} = \frac{r_i C}{R} \ . \tag{9}$$

Taking into account this analysis, we propose to assign to each trunk the following value of α:

$$\alpha_i = \frac{r_i B}{C} \ . \tag{10}$$

Computing α thresholds in this manner gives to each trunk its proportional share of the available bandwidth as desired but, in addition, this value is completely independent of the changing number and features of competing trunks.

4 Simulation Configuration

We have implemented *Ping Trunking* in the ns-2 simulator [14]. Figure 2 shows the network topology employed. It consists of three edge nodes and one core node belonging to a particular domain. The edge nodes E1 and E2 are connected to several TCP traffic sources whereas TCP sinks are connected to the edge node E3. We consider two competing aggregates: aggregate 1 comprises all the traffic that flows from E1 to E3, and aggregate 2 comprises all the traffic between E2 and E3. Therefore, both aggregates pass through a single bottleneck (link C-E3). Each aggregate consists of 50 TCP flows. All TCP connections established are modeled as eager FTP flows that always have data to send and last for the entire simulation time. We use the TCP New Reno implementation [15] and the size of data packets is set to 1 000 bytes.

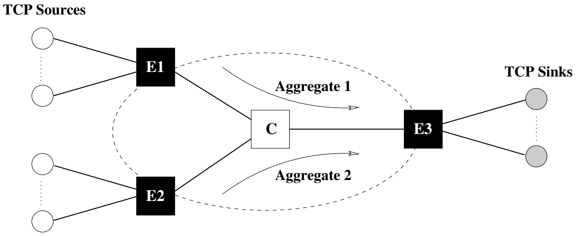


Fig. 2. Network topology. Every link has a 100 Mbps capacity and a propagation delay of 1 ms.

We consider two different scenarios. The first one represents a DiffServ domain with the two competing aggregates belonging to the same AF class. The marking scheme used in the edge routers is TSWTCM. The core node implements the RIO scheme: three sets of RED thresholds are maintained, one for each drop precedence. For the configuration of RIO parameters, the staggered setting [16] has been selected. The drop threshold values are shown in Fig. 3. The physical queue is limited to 150 packets and w_q equals 0.002.

In the second scenario, instead of DiffServ facilities, we employ *Ping Trunking* to regulate user data transmission. In this case, a trunk is used to manage each aggregate traffic stream. Each trunk buffer is a simple FIFO queue with capacity for 25 packets. The core queue is also a FIFO buffer limited to 150 packets. Control connections send 48-byte packets.[2] The maximum total backlog B is fixed to 50 packets.

Simulations run for 50 seconds. Each simulation experiment is repeated 10 times changing slightly the initial transmission time of each TCP flow and then, an average of the measured parameter and a 95% confidence interval for the

[2] Control connections do not actually transmit any real data, so control packets only consist of the TCP/IP header plus the overhead of the timestamp option required to estimate RTTs.

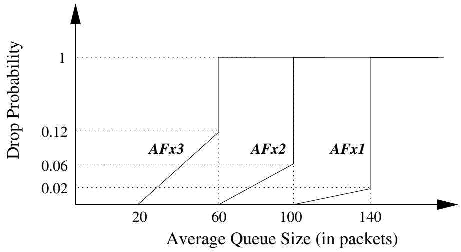


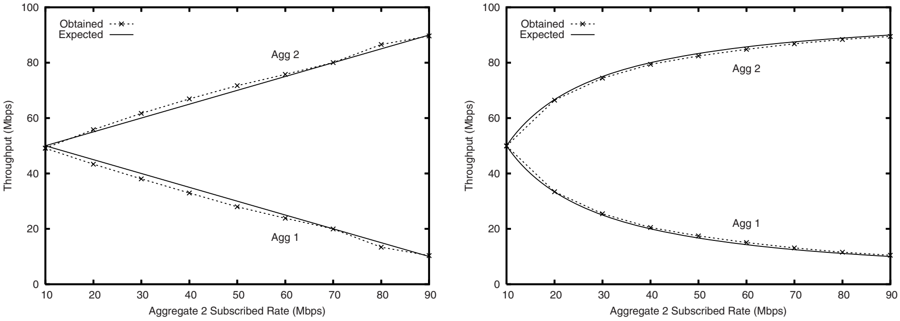
Fig. 3. Core RIO parameters in the AF scenario

mean value are taken over all runs. In any case, confidence intervals will not be represented in the graphs because they are lower than ±1 %.

5 Experimental Results

5.1 Bandwidth Distribution

In this experiment, we evaluate the effectiveness of both mechanisms when sharing the network bandwidth. We consider that aggregate 1 contracts a fixed throughput of 10 Mbps while the subscribed target rate of aggregate 2 varies from 10 to 90 Mbps. In the DiffServ scenario, the CIR value of each aggregate is set to its corresponding subscribed rate and the PIR value is set to the CIR value plus 10 Mbps. Figure 4 shows the throughput obtained by each aggregate with both techniques. The proportional share of bandwidth that should be assigned to each aggregate is also shown. With AF, there is an even distribution of excess bandwidth irrespective of the subscribed rates. In contrast, *Ping Trunking* divides the excess bandwidth in proportion to the subscribed rates. Though both solutions are acceptable, we consider it is more desirable that users with higher target rates obtain higher shares of excess bandwidth since target rates depend on the price that users pay.



(a) AF scenario

(b) Ping Trunking scenario

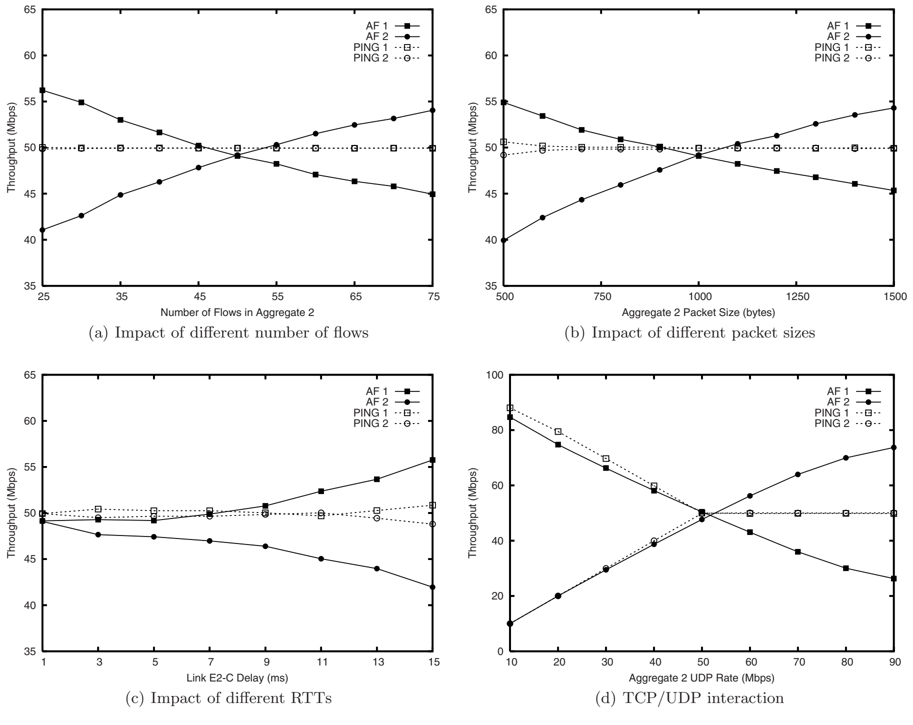
Fig. 4. Bandwidth distribution experiment results

5.2 Fairness Evaluation

In the previous simulations, both AF and *Ping Trunking* mechanisms have dealt with homogeneous aggregates, but, unfortunately, this is not always the usual scenario. Aggregates can be composed of different numbers of flows, can send packets of different sizes and can have different RTTs. In addition, aggregates can carry UDP flows, which are not congestion aware. In this section, we have conducted several simulations to verify that our proposal can be used to share the bandwidth among heterogeneous aggregates in a fair manner. We consider that the two competing aggregates have contracted the same target rate (10 Mbps).

The first key factor studied has been the number of flows in the competing aggregates. In this experiment, we consider that aggregate 2 contains a different number of TCP flows, varying from 25 to 75. Figure 5(a) shows the obtained results. In the AF case, the aggregate with a larger number of TCP flows obtains a greater share of the available bandwidth. However, with our proposal, each aggregate obtains an equal amount.

Fairness is also desired between aggregates carrying packets of different sizes. We have simulated a second experiment where aggregate 2 packet size increases from 500 to 1500 bytes. The results are shown in Fig. 5(b). Through AF, the aggregate that is sending larger packets consumes more of the available



(a) Impact of different number of flows

(b) Impact of different packet sizes

(c) Impact of different RTTs

(d) TCP/UDP interaction

Fig. 5. Fairness evaluation experiment results

bandwidth. Under *Ping Trunking*, the sharing of bandwidth can be made insensitive to packet sizes.

Another important factor in the share of bandwidth is the RTT of the competing aggregates. In order to compare the throughput obtained by aggregates with different RTTs, we consider that the delay of the link that joins edge node E2 with the core node has been changed from 1 to 10 ms. As showed in Fig. 5(c), through AF, aggregates with different RTTs cannot achieve a fair share of bandwidth and the shorter the RTT, the higher the obtained throughput. In contrast, *Ping Trunking* is able to amend such unfairness giving to each aggregate its fair share.

Finally, it is important to protect responsive TCP flows from non-responsive UDP flows since this unresponsive traffic may impact the TCP traffic adversely. In this last experiment, aggregate 1 contains 50 TCP flows, while aggregate 2 has a single UDP flow with a sending rate increasing from 10 to 90 Mbps. Figure 5(d) shows the obtained results. Under the AF case, as the UDP rate increases, the amount of bandwidth obtained by the TCP aggregate decreases. With our proposal, this unfairness problem is absent and the bandwidth can be shared in a TCP-friendly manner.

6 Related Work

Many smart packet marking mechanisms have been proposed to overcome these unfairness issues found in AF. *Adaptive Packet Marking* [17] is one of these schemes able to provide soft bandwidth guarantees, but it has to be implemented inside the TCP code itself and thus, requires varying all TCP agents. Intelligent traffic conditioners proposed in [18] handle a subset of these fairness problems using a simple TCP model when marking packets. However, these conditioners require external inputs and cooperation among markers for different aggregates complicating both implementation and deployment. Another marking algorithm based on a more complex TCP model is *Equation-Based Marking* [19]. This scheme solves the fairness problems associated with heterogeneous TCP flows under diverse network conditions. Its behavior depends on the quality of the estimation of the current loss rate seen by TCP flows. Unfortunately, the calculation of this estimate is not an easy problem and complicates the deployment of the scheme extremely. In [20], an RTT-RTO aware conditioner is proposed, but this scheme only mitigates RTT bias. The *Counters-Based Modified* traffic conditioner [21] is able to cope with TCP flows with variable target rates and RTTs, but it cannot oversee UDP traffic. In [22], an adaptive token bucket algorithm able to provide each aggregate with its fair share of the available bandwidth in proportion to the target rate is presented. This approach is based on edge-to-edge feedback information conveyed in TCP acknowledgements, so it cannot be used to manage aggregates just containing UDP flows exclusively.

A different approach addresses these problems by enhanced RIO queue management algorithms. Examples of this technique are DRIO [23] and DAIO [24] schemes. Both techniques require maintaining state information of each individual flow at core routers. Since there can be thousands of active flows, these solutions need to store and manage a great amount of state information at the core of the network and therefore, they are not scalable. Other enhanced RIO algorithms such as EDRIO [25] and URIO [26] use proper buffer usage policing at the aggregate level to avoid scalability issues, but the need to adjust all the core routers still hinders their deployment.

7 Conclusions and Future Work

Both AF PHB and *Ping Trunking* mechanisms can be used to provide assured services to network users. However, the edge-to-edge management carried out by our proposal provides this service without the need for modifying core nodes. This feature is very interesting because it facilitates the deployment of our proposal substantially. In addition, our proposal guarantees the required fairness on bandwidth sharing among heterogeneous aggregate traffic streams.

In future work we plan to take advantage of *Ping Trunking* features to prioritize user traffic strictly based on the application type. For example, users could mark packets from highly interactive applications, such as Telnet or Web browsing, with a high priority, and packets from less interactive applications, such as FTP, with a lower one. Thus, if congestion occurs, low priority packets could be dropped more frequently at the trunk buffer using a suitable active queue mechanism. Following this approach, we will be in a position to fairly distribute the network bandwidth among competing aggregates while protecting interactive applications at the same time.

References

1. Heinanen, J., Baker, F., Weiss, W., Wroclawski, J.: Assured Forwarding PHB group. RFC 2597 (1999)
2. Blake, S., Black, D., Carlson, M., Davis, E., Wang, Z., Weiss, W.: An architecture for differentiated services. RFC 2475 (1998)
3. Herrería-Alonso, S., Fernández-Veiga, M., Rodríguez-Pérez, M., Suárez-González, A., López-García, C.: Ping Trunking: A Vegas-like congestion control mechanism for aggregated traffic. Lecture Notes in Computer Science (QofIS'04) **3266** (2004) 104–113
4. de Rezende, J.F.: Assured service evaluation. In: Proceedings of IEEE GLOBECOM. (1999) 100–104
5. Seddigh, N., Nandy, B., Pieda, P.: Bandwidth assurance issues for TCP flows in a differentiated services network. In: Proceedings of IEEE GLOBECOM. (1999) 1792–1798
6. Fang, W., Seddigh, N., Nandy, B.: A time sliding window three colour marker (TSWTCM). RFC 2859 (2000)
7. Clark, D.D., Fang, W.: Explicit allocation of best effort packet delivery. IEEE/ACM Transactions on Networking **6** (1998) 362–373
8. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking **1** (1998) 397–413

9. Chapman, A., Kung, H.T.: Traffic management for aggregate IP streams. In: Proceedings of 3rd Canadian Conference on Broadband Research. (1999) 1–9
10. Kung, H.T., Wang, S.Y.: TCP Trunking: Design, implementation and performance. In: Proceedings of 7th Int. Conference on Network Protocols. (1999) 222–231
11. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol label switching architecture. RFC 3031 (2001)
12. Brakmo, L., O'Malley, S., Peterson, L.: TCP Vegas: New techniques for congestion detection and avoidance. In: Proceedings of ACM SIGCOMM. (1994) 24–35
13. Low, S., Peterson, L., Wang, L.: Understanding Vegas: A duality model. In: Proc. of ACM SIGMETRICS. (2001)
14. ns-2.27: The network simulator (2004)
15. Fall, K., Floyd, S.: Simulation-based comparison of Tahoe, Reno, and Sack TCP. Computer Communication Review **26** (1996) 5–21
16. Makkar, R., Lambadaris, I., Salim, J.H., Seddigh, N., Nandy, B., Babiarz, J.: Empirical study of buffer management schemes for diffserv assured forwarding PHB. Technical report, Nortel Networks (2000)
17. Feng, W., Kandlur, D., Saha, D., Shin, K.: Adaptive packet marking for maintaining end-to-end throughput in a differentiated-services internet. IEEE/ACM Transactions on Networking **7** (1999) 685–697
18. Nandy, B., Seddigh, N., Pieda, P., Ethridge, J.: Intelligent traffic conditioners for assured forwarding based differentiated services networks. Lecture Notes in Computer Science (Networking'00) **1815** (2000) 540–554
19. El-Gendy, M., Shin, K.: Equation-based packet marking for assured forwarding services. In: Proceedings of IEEE INFOCOM. (2002) 845–854
20. Habib, A., Bhargava, B., Fahmy, S.: A round trip time and time-out aware traffic conditioner for differentiated services networks. In: Proceedings of IEEE International Conference on Communications (ICC). (2002) 981–985
21. Cano, M.D., Cerdan, F., Garcia-Haro, J., Malgosa-Sanahuja, J.: Performance analysis of the counters-based modified traffic conditioner in a diffserv network. In: Proceedings of IEEE International Symposium on Computers and Communications (ISCC). (2003) 305–311
22. Park, E.C., Choi, C.H.: Proportional bandwidth allocation in diffserv networks. In: Proceedings of IEEE INFOCOM. (2004) 2039–2050
23. Lin, W., Zheng, R., Hou, J.: How to make assured services more assured. In: Proceedings of ICNP. (1999)
24. Su, L., Hou, J.: An active queue management scheme for internet congestion control and its application to differentiated services. In: Proceedings of IEEE ICCCN. (2000) 62–68
25. Herrería-Alonso, S., Fernández-Veiga, M., López-García, C., Rodríguez-Pérez, M., Suárez-González, A.: Improving fairness requirements for assured services in a differentiated services network. In: Proceedings of IEEE International Conference on Communications (ICC). (2004) 2076–2080
26. Herrería-Alonso, S., Rodríguez-Pérez, M., Fernández-Veiga, M., Suárez-González, A., López-García, C.: Unbiased RIO: An active queue management scheme for diffserv networks. Lecture Notes in Computer Science (ECUMN'04) **3262** (2004) 60–69

Max-Min Fair Distribution of Modular Network Flows on Fixed Paths

Pål Nilsson[1] and Michał Pióro[1,2]

[1] Department of Communication Systems, Lund University, Sweden,
Box 118 SE-221 00, Lund, Sweden
{paln, mpp}@telecom.lth.se
[2] Institute of Telecommunications, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warszawa, Poland

Abstract. In this paper a new aspect of the classical max-min fairness fixed-path problem is investigated. The considered (multi-criteria) optimization problem is almost identical to the continuous-flow problem, with the additional complicating assumption that flows must be integral. We show that such an extension makes the problem substantially more difficult (in fact $\mathcal{NP}$-hard). Through comparison with the closely related continuous-flow problem, a number of properties for the solution of the extended problem are derived. An algorithm, based on sequential resolution of linear programs, that shows to be useful (produce optimal solutions) for many instances of the considered problem, is given. It follows that this algorithm can be made exact, through substituting the involved linear programs by mixed-integer programs.

Keywords: Max-min fairness, Network optimization, Modular flows.

1 Introduction

This paper concerns Max-Min Fair (MMF) allocation of bandwidth to demands (users) in a communication network. The MMF allocation principle is often considered in the context of IP networks carrying elastic traffic. It is also relevant when several local networks are to be connected via an overlay network with given overlay link capacities. In such a situation it has to be decided how much bandwidth to assign to the pairs of these local networks.

We consider the case when one (fixed) path per demand is used, and when demands can be assigned bandwidth only in multiples of a predefined module. This is a practical extension of the frequently cited MMF problem addressed in [1]. The modular (integral) flow requirement has to the best our knowledge not been studied in this context before. However, a great deal of work has been carried out considering other versions of this problem. In [1] it is shown how MMF is obtained if paths are fixed and demand flows are continuous. It is also well-known how MMF can be achieved if flows are allowed to split over several paths and demand flows are continuous [2, 3, 4]. The hardness of computing MMF continuous flows, forcing single-path selection, was pointed out in [5, 6]. Integral flow volumes, certainly being the source of difficulty for the problem

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 916–927, 2006.

considered in this paper, can be well motivated from a practical viewpoint. This requirement models that each demand volume must be a multiple of a predefined module, and is a consequence of that in a real network there is always a smallest trading unit, prohibiting flows from being continuous. It will be assumed that a problem instance is given as a network with link capacities, a set of source-destination node-pairs (S-D pairs), where each S-D pair represents a requirement for bandwidth (demand), and a path for each demand. For such an instance, the following traffic engineering problem is addressed: the demand between each S-D pair must be assigned a modular flow volume, such that the sum of flows on each link does not violate the link capacity. The distribution of flows among the S-D pairs must obey the MMF principle.

1.1 Notation

Throughout the paper we will use the following notation. Vertices (nodes) are labeled with index v, where $v = 1, 2, \ldots, V$, and V is the number of vertices. Vertices are interconnected by edges (links) labeled with index e, where $e = 1, 2, \ldots, E$, and E is the number of edges. Edges are assumed to be undirected. Each edge e is assigned a certain given capacity denoted by c_e. Demands are labeled with index d, where $d = 1, 2, \ldots, D$, and D is the number of demands. A demand is a requirement for bandwidth between a vertex-pair in the network. Each demand d is assumed to be associated with one selected simple path. A path is a set of edges that connects a vertex-pair. A binary indicator, δ_{ed}, is used for the edge-demand incidence relation: $\delta_{ed} = 1$ if edge e belongs to the path of demand d, and $\delta_{ed} = 0$ otherwise. The total flow allocated to demand d (on its corresponding path) will be identified by x_d. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$ denote the vector where entry number d is the flow x_d allocated to demand d (also called the allocation vector), and let $\vec{\boldsymbol{x}}$ be the allocation vector sorted in non-decreasing order. The sorted allocation vector, $\vec{\boldsymbol{x}}$, is said to be lexicographically greater than the sorted allocation vector, $\vec{\boldsymbol{x}}'$, $\vec{\boldsymbol{x}} \succ \vec{\boldsymbol{x}}'$, if the first non-zero entry of $\vec{\boldsymbol{x}} - \vec{\boldsymbol{x}}'$ is positive. Consequently, $\vec{\boldsymbol{x}} \succeq \vec{\boldsymbol{x}}'$ means $\vec{\boldsymbol{x}} \succ \vec{\boldsymbol{x}}'$ or $\vec{\boldsymbol{x}} = \vec{\boldsymbol{x}}'$.

1.2 Problem Description

This study concerns the problem of assigning modular flows to demands in a capacitated network, such that the distribution of flows among demands is MMF. The MMF principle is to first assure that the demand that gets the least flow gets as much as possible, then that the demand that gets the second least flow gets as much as possible, and so on. Formally, in an MMF allocation of flows each demand is assigned a flow such that it holds for the sorted allocation vector that an entry can be increased only at the cost of decreasing a previous entry, or by making the allocation vector infeasible[1]. It can be shown that obtaining an allocation vector, $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$, with this characteristic is equivalent to solving

$$\text{lex max } \vec{\boldsymbol{x}};\ \boldsymbol{x} \in Q, \tag{1}$$

[1] Since this is a property of the sorted allocation vector entries and not for the specific demands, the definition is valid even for non-convex versions of the problem.

where Q is the set of feasible solutions [4]. In other words, (1) seeks for an allocation vector, $\boldsymbol{x}^* \in Q$, for which it holds that $\vec{\boldsymbol{x}}^* \succeq \vec{\boldsymbol{x}}$ for all $\boldsymbol{x} \in Q$. In this paper, the set of feasible solutions Q is defined by the following two requirements:

(i) $\sum_d \delta_{ed} x_d \leq c_e$, $e = 1, 2, \ldots, E$, and
(ii) $x_d \in \mathbb{Z}^+$ for all demands $d = 1, 2, \ldots, D$,

where $\mathbb{Z}^+$ is the non-negative integers. The above two constraints mean in turn that the sum of flows on an edge cannot exceed the edge's capacity, and that each flow must assume a non-negative integer. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$, be a feasible solution to (1), with Q constituted only by (i), and let $\boldsymbol{x}^z = (x_1^z, x_2^z, \ldots, x_D^z)$, be a feasible solution to (1), with Q constituted by (i) and (ii). To simplify notation, let $\boldsymbol{y} = (y_1, y_2, \ldots, y_D) = \vec{\boldsymbol{x}}$ and $\boldsymbol{y}^z = (y_1^z, y_2^z, \ldots, y_D^z) = \vec{\boldsymbol{x}}^z$. We will denote optimal $\boldsymbol{x}$ an Optimal Continuous Solution (OCS), and optimal $\boldsymbol{x}^z$ an Optimal Integral Solution (OIS). Note that solving for OCS is precisely what is addressed in [1], and is well known to be accomplished by a simple algorithm (called "lifting" or "waterfilling") of polynomial time [1, 4].

Example 1. Consider the network given in Figure 1. One demand between each vertex-pair is assumed. Paths are evident. Edge capacities are given in the figure. The sorted OCS is $\boldsymbol{y} = (0.5, 0.5, 0.5)$, whereas the sorted OIS is $\boldsymbol{y}^z = (0, 1, 1)$.

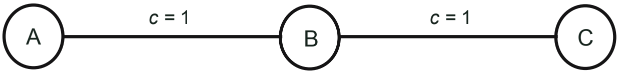


Fig. 1. A simple instance

1.3 The Assumption of Modular Flows

In practice, an edge cannot have an arbitrary capacity, but is installed in multiples of a predefined module, M. Moreover, it is reasonable to assume that for each demand d, $d = 1, 2, \ldots, d$, the demand's flow, x_d, must be a multiple of the same module. Without loss of generality we may, just changing units, assume that $M = 1$, and thus that $c_e \in \mathbb{Z}^+ \setminus \{0\}$, and, as is accomplished by (ii), require that $x_d \in \mathbb{Z}^+$ for all demands $d = 1, 2, \ldots, D$. Hence, modular flows can be treated as integral flows.

2 Some Properties of the Optimal Integral Solution

As the properties of the optimal continuous solution are very well known and understood [1, 3, 4], we will in this section address characterization of the optimal integral solution by comparing it to the optimal continuous solution. We assume that $\boldsymbol{y} = \vec{\boldsymbol{x}}$ is the sorted OCS, and $\boldsymbol{y}^z = \vec{\boldsymbol{x}}^z$ is the sorted OIS.

Property 1. If $\boldsymbol{y}^z \neq \boldsymbol{y}$ then there exists an entry k for which $y^z_k > y_k$.

Proof. It is easy to see that $\boldsymbol{y} \neq \boldsymbol{y}^z$ implies that there exists a demand d such that $x_d - \lfloor x_d \rfloor > 0$, because if $x_d \in \mathbb{Z}^+$, for all $d = 1, 2, \ldots, D$, then the OCS would be an OIS and necessarily $\boldsymbol{y} = \boldsymbol{y}^z$. Without loss of generality we may assume that $\boldsymbol{x} = \boldsymbol{y}$ (this may be obtained by alternative enumeration of demands). Consider the non-integral entries $x_i, x_{i+1}, \ldots, x_{i+m}$ for which it holds that for all entries k, $k < i$, (if any) $x_k \in \mathbb{Z}^+$, and if there exists an entry $i+m+1$ then $x_{i+m+1} \in \mathbb{Z}^+$. Construct a solution $\boldsymbol{x}^a$ by truncating all demand volumes of $\boldsymbol{x}$, except demand $i+m$, which is rounded up. The idea is illustrated in Figure 2. This solution must be feasible since edge capacities are integral. Moreover $\boldsymbol{x}^a$ is an integral solution with the property that $\boldsymbol{x}^a = \vec{\boldsymbol{x}}^a (= \boldsymbol{y}^a)$. As it must hold that $\boldsymbol{y}^z \succeq \boldsymbol{y}^a$, we must have that either $y^z_j > y^a_j$ for some j, $i \leq j < m$, or that $y^z_j = y^a_j$ for all j, $i \leq j < m$ and that $y^z_{i+m} \geq y^a_{i+m}$. Both cases imply that there exists an index j, $i \leq j \leq i+m$, such that $y^z_j > y_j$. □

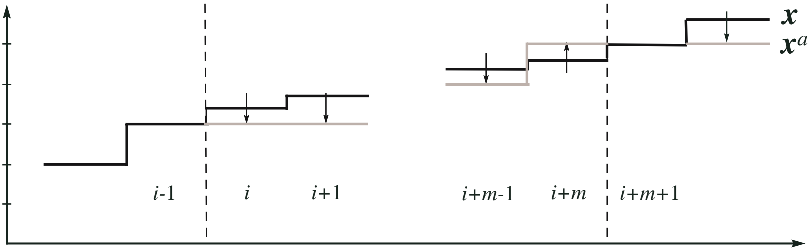


Fig. 2. How to obtain $\boldsymbol{x}^a$ from $\boldsymbol{x}$

Property 2. If for some entry k, $y^z_k > y_k$, then there exists a demand d, such that $x^z_d > x_d$.

Proof. Without loss of generality assume that $\boldsymbol{x} = \boldsymbol{y}$ (if this is not true, reenumerate the demands). Suppose $\boldsymbol{x} \geq \boldsymbol{x}^z$ and consider all entries m and n, $1 \leq m < n \leq D$, such that $x^z_n < x^z_m$. Interchanging all such elements in $\boldsymbol{x}^z$, we will eventually arrive at $\boldsymbol{y}^z$. However, we have that $x_m \geq x^z_m > x^z_n$ and $x_n \geq x_m \geq x^z_m$, so it must be true that $\boldsymbol{x} = \boldsymbol{y} \geq \boldsymbol{y}^z$, which is a contradiction. □

Property 3. Let j be the largest integer for which it is true that $y_k - y^z_k \geq 0$, $1 \leq k \leq j$. Then, $j \geq 1$ and it holds that $y_k - y^z_k < 1$, for $1 \leq k \leq j$.

Proof. By definition such an entry j must exist. Suppose that for some k, $1 \leq k \leq j$, $y_k - y^z_k \geq 1$. Then we can find a feasible integral solution by just truncating the OCS. Call this solution $\boldsymbol{y}^t$. Apparently, $\boldsymbol{y}^t \succ \boldsymbol{y}^z$, which is a contradiction proving the second part of the statement. □

The following property is a direct analogy to a very well known property of the OCS [1, 4].

Property 4. For each demand d', there exists at least one saturated edge e for which $x^z_{d'} \geq \max_d\{x^z_d : \delta_{ed} = 1\} - 1$.

Proof. It is obvious that it is possible to find at least one saturated edge for each demand, since otherwise that demand could be increased. Denote by

$$\hat{x}^z_e = \max_d\{x^z_d : \delta_{ed} = 1\},$$

the flow of the maximal demand on edge e, and suppose, contradictory to the statement, that it holds for all such saturated edges e, for demand d', that $x^z_{d'} < \hat{x}^z_e - 1$. Then, for these saturated edges we have $\hat{x}^z_e > x^z_{d'} + 1$. Thus for each of the edges with this property, reassigning the currently maximal flow a value of $\hat{x}^z_e - 1$ and demand d' a value of $x^z_{d'} + 1$ give a lexicographically larger solution, which is a contradiction. □

It should be noted that Property 4 implies that for each demand d', if there does not exist a saturated edge for which $x^z_{d'} = \max_d\{x^z_d : \delta_{ed} = 1\}$, then there must exist a saturated edge for which $x^z_{d'} = \max_d\{x^z_d : \delta_{ed} = 1\} - 1$.

Property 5. For a feasible allocation vector $\boldsymbol{x}^m$, $\boldsymbol{x}^m \in (\mathbb{Z}^+)^D$, if it holds for each demand d', $d' = 1, 2, \ldots, D$, that $x^m_{d'} = \max_d\{x^m_d : \delta_{ed} = 1\}$ on at least one saturated link e for which $\delta_{ed'} = 1$, then $\boldsymbol{x}^m$ is the unique OIS.

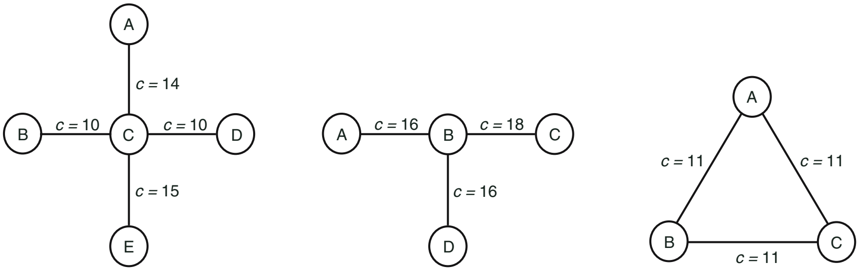
Proof. The result is valid for the continuous flows case [4], i.e., a feasible allocation vector, $\boldsymbol{x}$, for which it holds that for each demand d', $x_{d'} = \max_d\{x_d : \delta_{ed} = 1\}$ on at least one saturated link e for which $\delta_{ed'} = 1$, is the unique OCS. Now since $\boldsymbol{x}^m \in (\mathbb{R}^+)^D$, $\boldsymbol{x}^m$ is the unique OCS and therefore the unique OIS. □

The following examples illustrate that it may happen, considering the OCS and the OIS for a given instance, that $x^z_d < \lfloor x_d \rfloor$ and that $x^z_d > \lceil x_d \rceil$. They also show that there is no certain throughput domination, i.e., there exist both instances for which $\sum_d x^z_d < \sum_d x_d$, and instances for which $\sum_d x^z_d > \sum_d x_d$.

Example 2. Consider the network given in Figure 3(a). There are 2 demands between verticess A and B, 2 between A and E, 2 between A and D, 1 between C and B, 1 between C and E, and 1 between C and D. The sorted OCS is $\boldsymbol{y} = (\underbrace{7/3, \ldots, 7/3}_{6}, 16/3, 16/3, 31/3)$ and the sorted OIS is $\boldsymbol{y}^z = (2, 2, 2, 2, 3, 3, 6, 6, 9)$.

Example 3. Consider the network given in Figure 3(b). There are 4 demands between verticess A and B, 2 between A and C, 4 between D and B, 2 between D and C, and 1 between D and B. The sorted OCS is $\boldsymbol{y} = (\underbrace{8/3, \ldots, 8/3}_{12}, 22/3)$ and the sorted OIS is $\boldsymbol{y}^z = (2, 2, 2, 2, \underbrace{3, \ldots, 3}_{8}, 10)$.

Example 4. Consider the network given in Figure 3(c). Edge capacities are given in the figure. There is one demand between each vertex-pair. Each demand is using the associated simple two-edge path. The sorted OCS is $\boldsymbol{y} = (5.5, 5.5, 5.5)$ and the sorted OIS is $\boldsymbol{y}^z = (5, 5, 6)$.



(a) Example 2 (b) Example 3 (c) Example 4

Fig. 3. Example instances comparing the OCS and the OIS

3 Computational Complexity

In this section it will be shown that the problem studied in this paper is $\mathcal{NP}$-hard. This means that it is unrealistic to aim for a general polynomial time algorithm that obtains an MMF flow distribution, when integer flows on fixed paths are required. Mind that the considered optimization problem is exactly that considered in [1], but with integer-valued flows. As can be verified in Examples 2 and 3, the basic "lifting" algorithm (sometimes called "waterfilling") presented in [1] is in general not applicable for the integer flows case. An attempt to use this basic procedure will show that certain non-trivial, discrete decisions occasionaly have to be taken. So there are certainly reasons to conjecture that this multi-criteria optimization problem is computationally hard. We will call the associated decision problem FIXED PATHS MMF – MODULAR FLOWS (FIXMMF-MF):

FIXMMF-MF:
INSTANCE: An edge capacity $c_e \in \mathbb{Z}^+$ for each edge $e = 1, 2, \ldots, E$, a binary edge-demand incidence coefficient, δ_{ed}, for each demand $d = 1, 2, \ldots, D$, and a target vector $\boldsymbol{x}^T \in (\mathbb{Z}^+)^D$.
QUESTION: Is there an assignment of flow, $x_d \in \mathbb{Z}^+$, for each demand d, such that $\sum_d \delta_{ed} x_d \leq c_e$ for each edge e, and such that if $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$, then $\vec{\boldsymbol{x}} \succeq \vec{\boldsymbol{x}}^T$?

Proposition 1. *FIXMMF-MF is $\mathcal{NP}$-complete.*

Proof. A nondeterministic algorithm needs only to guess an integral flow for each demand and check if the edges have the required capacity and if it holds for the resulting allocation vector, $\boldsymbol{x}$ that $\vec{\boldsymbol{x}} \succeq \vec{\boldsymbol{x}}^T$. Thus clearly, FIXMMF-MF is in $\mathcal{NP}$. We will transform the decision problem of SET PACKING into an instance of FIXMMF-MF. It is trivial to verify $\mathcal{NP}$-completeness of the former, restricting it to EXACT COVER BY 3-SETS, shown to be $\mathcal{NP}$-complete in [7].

SET PACKING:
INSTANCE: A collection of C finite sets and a positive integer $K \leq |C|$.
QUESTION: Does C contain at least K mutually disjoint sets?

Consider an arbitrary instance of SET PACKING. A collection C of n finite sets is given, $C = \{A_1, A_2, \ldots, A_n\}$. We will let each set A_i constitute a demand and the elements of each such set a chain of edges that is a path for that demand. Assume that there is N distinct elements in total in all of the sets A_i. For each such element a_k, $k = 1, 2, \ldots, N$, construct two vertices connected by one edge as in Figure 4. These edges will be referred to as the element edges. For each set $A_i \in C$ perform the following operations. Construct a source vertex s_i and a sink vertex t_i. Label the elements of set A_i such that $A_i = \{z_1, z_2, \ldots, z_m\}$, and note that there is one-to-one correspondence between these labeled elements and m of the element edges. Denote the upper vertex of the element edge corresponding to z_j, $j = 1, 2, \ldots, m$ by v_j^u, and the lower vertex by v_j^l. Add new edges connecting s_i to v_1^u, v_1^l to v_2^u, v_2^l to v_3^u, and so on. Finally, add an edge that connects v_m^l with the sink vertex t_i. This constitutes a path between s_i and t_i, traversing all element edges of A_i. Note that all edges on this path that are not element edges can only be used by vertex-pair (demand) (s_i, t_i). Assign a capacity of 1 to all edges. Let $\boldsymbol{x}^T = (\underbrace{0, \ldots, 0}_{n-K}, \underbrace{1, \ldots, 1}_{K})$. This constitutes an instance of FIXMMF-MF, with n demands. Suppose that we have a positive answer to SET PACKING. This implies that there exist at least K mutually disjoint sets A_i. Assigning a flow of 1 to each of the corresponding vertex-pairs (s_i, t_i), and 0 to the rest will give $\vec{\boldsymbol{x}} = (\underbrace{0, \ldots, 0}_{n-H}, \underbrace{1, \ldots, 1}_{H})$, $H \geq K$. Thus $\vec{\boldsymbol{x}} \succeq \vec{\boldsymbol{x}}^T$, and we have a positive answer to FIXMMF-MF. Contrarily, suppose that we have a positive answer to FIXMMF-MF, i.e., that there exists a feasible allocation $\boldsymbol{x}$, with $\vec{\boldsymbol{x}} \succeq (\underbrace{0, \ldots, 0}_{n-K}, \underbrace{1, \ldots, 1}_{K})$. Since capacities are equal to 1, no edges can be shared by demands and $x_d \leq 1$ for all $d = 1, 2, \ldots, D$. This implies that paths of demands that are assigned flow 1 must be disjoint. As constructed, these paths define at least K mutually disjoint sets A_i, and a positive answer to SET PACKING follows. Hence, since SET PACKING is $\mathcal{NP}$-complete, FIXMMF-MF is $\mathcal{NP}$-complete. □

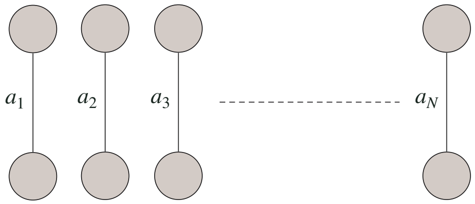


Fig. 4. Element edges

Knowledge of the lifting algorithm and a bit of reflection reveals that the above result will not hold (unless $\mathcal{P} = \mathcal{NP}$) if $\boldsymbol{x}^T = c \cdot \boldsymbol{e}$, where $\boldsymbol{e}$ is the unity vector of size D, and c is a positive integer (corresponding to the optimization problem of finding the maximal first entry of the sorted allocation vector) nor if each demand's path has at most one shared link, both cases for which modified versions of the lifting algorithm solves the problems in polynomial time [8].

4 An Algorithm

In this section we present an algorithm for the considered problem. The approach is in essence exploiting the "distribution approach" described for non-convex MMF problems in general in [9]. The distribution approach makes use of that if an MMF problem only has a discrete (finite) set of possible outcome values, then it can be stated as a lexicographic minimization. Even though the presented algorithm does not offer any general optimality or running time guarantees, we show that if the algorithm terminates with an integral solution, it must be optimal for the considered problem. In the following section this will be shown to happen quite frequently. If this is not the case, it is possible to modify the algorithm (although with the risk of making it heavily computationally complex) such that the output is guaranteed to be an optimal integral solution. The considered algorithm consists in successive resolution of Linear Programs (LPs). As there exist efficient methods for solving LPs (it will become evident that the algorithm solves at most r LPs with at most $D + rD$ variables and at most $rD + (r-1) + E$ constraints, where $r \le \max_e\{c_e\}$) the algorithm may be an appealing way of approaching an instance of the problem. Define the linear programming problem P_k,

$$P_k : \qquad \tau_k = \min \sum_d t_{kd} \tag{2}$$

$$\text{s.t.} \quad l - x_d \le t_{ld}, \; l = 1, 2, \ldots, k, \; d = 1, 2, \ldots, D \tag{3}$$

$$\sum_d t_{ld} \le \lceil \tau_l \rceil, \; l = 1, 2, \ldots, k-1 \tag{4}$$

$$\sum_d \delta_{ed} x_d \le c_e, \; e = 1, 2, \ldots, E \tag{5}$$

$$x_d, t_{ld} \ge 0, \tag{6}$$

and consider Algorithm 1.

Proposition 2. *Let $\boldsymbol{x}'$ be the output of Algorithm 1. If $\boldsymbol{x}' \in (\mathbb{Z}^+)^D$, then $\boldsymbol{x}' \in Q$, and $\boldsymbol{x}'$ is an optimal solution to the considered problem, i.e., $\vec{\boldsymbol{x}}' =$* lex max $\{\vec{\boldsymbol{x}} : \boldsymbol{x} \in Q\}$*, where Q is the constraint set of (i) and (ii).*

Proof. Note that we may assign $r = \max_e\{c_e\}$, as for sure it must hold that $x_d \le \max_e\{c_e\}$ for all d, $d = 1, 2, \ldots, D$. Note further that Proposition 2 does *not* require that solutions to intermediate LPs (P_k, $k < r$) are integral.

Algorithm 1.

```
$k := 1$, improvement:=TRUE, $r := \max_e\{c_e\}$
while improvement and $k \le r$ do
   solve problem $P_k$ for $\boldsymbol{x}^*$ and $\tau_k$
   if $\max\{\boldsymbol{x}^*\} < k$ then
      improvement:=FALSE
   end if
   $k := k + 1$
end while
$\boldsymbol{x}' := \boldsymbol{x}^*$
```

We will need some additional notation. As earlier we will use $\boldsymbol{y}$ to denote a sorted (in non-decreasing order) version of allocation vector $\boldsymbol{x}$, i.e, $\boldsymbol{y} = \vec{\boldsymbol{x}}$. Let $f : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ be the function for which $f(l,x) = \begin{cases} l - x & \text{if } l \ge x \\ 0 & \text{if } l < x \end{cases}$. Let $\boldsymbol{x}^i = (x_1^i, x_2^i, \ldots, x_D^i)$, $1 \le i \le r$, be a solution to

$$\text{lex max } \vec{\boldsymbol{x}} \tag{7}$$

$$\text{s.t.} \sum_d \delta_{ed} x_d \le c_e,\ e = 1,2,\ldots,E \tag{8}$$

$$x_d \in \{0,1,\ldots,i\},\ d = 1,2,\ldots,D, \tag{9}$$

and note that it always holds that $\sum_d f(i, x_d^i) = \sum_d f(i, x_d^j)$, for any (i,j) such that $1 \le i < j \le r$. For the proof we will assume that that the algorithm terminates after r iterations, since if it halts after u $(u < r)$ iterations, we may redefine r as $r := u$. As the solution in that case does not improve for $k = u+1$, it follows that the (sorted) allocation vector being the solution to problem P_{u+j} cannot be lexicogaphically greater than the sorted solution to problem P_u, for any integer $j \ge 1$. Without loss of generality we may also assume that for a solution $\boldsymbol{x}^*$ to P_k it holds that $x_d^* \le k$, for all $d = 1,2,\ldots,D$, since if $\exists d : x_d^* > k$, we may assign $x_d^* = k$ for all such d, maintaining feasibility and objective function value. Further it must hold for $\boldsymbol{x}'$ that $\sum_d f(l, x_d') = \lceil \tau_l \rceil$, for all l, $1 \le l \le r$. For $l = r$ this follows directly from the fact that $\tau_r = \lceil \tau_r \rceil$ (as $\boldsymbol{x}' \in (\mathbb{Z}^+)^D$) and that $\tau_r = \sum_d f(r, x_d')$ by definition. For $l < r$, suppose that $\sum_d f(l, x_d') > \lceil \tau_l \rceil$. Then $\boldsymbol{x}'$ is infeasible for P_r, which is a contradiction. On the other hand, suppose that i, $i < r$, is the smallest positive integer for which $\sum_d f(i, x_d') < \lceil \tau_i \rceil$, holds. As $\boldsymbol{x}' \in (\mathbb{Z}^+)^D$, $\sum_d f(i, x_d') \in \mathbb{Z}^+$ and $\sum_d f(i, x_d') < \tau_i$ must be true, contradicting that τ_i is the optimal objective function value for P_i.

Let $\boldsymbol{x}^*$ be an optimal solution to P_1. We have that $\tau_1 = \sum_d f(1, x_d^*)$ and $\lceil \tau_1 \rceil = \sum_d f(1, x_d')$, so $0 \le \sum_d f(1, x_d') - \sum_d f(1, x_d^*) < 1$. Now suppose that $\boldsymbol{y}^1 \succ \boldsymbol{y}'$. Then $\sum_d f(1, x_d') - \sum_d f(1, x_d^1) \ge 1$ and thus $\sum_d f(1, x_d^1) < \sum_d f(1, x_d^*)$, which is a contradiction. Hence $\boldsymbol{y}' \succeq \boldsymbol{y}^1$.

Assume that $\boldsymbol{y}' \succeq \boldsymbol{y}^{k-1}$. Then $\boldsymbol{y}' \succeq \boldsymbol{y}^i$, $1 \le i \le k-1$. Let m be the number of entries of $\boldsymbol{x}^{k-1}$ that are strictly smaller than $k-1$. We will start by proving that $y_j^k = y_j'$ for $j \le m$. Again aiming for contradiction, suppose that entry p,

$p \leq m$, is the first entry for which $y^k_p \neq y'_p$. Then either $y^k_p < y'_p$ or $y^k_p > y'_p$. Suppose $y^k_p < y'_p = t$. This facilitates derivation of a solution $\boldsymbol{x}''$ from $\boldsymbol{x}'$ by assigning $x''_d = x'_d$ if $x'_d \leq t$ and $x''_d = t$ if $x'_d > t$. It will hold that $\sum_d \delta_{ed} x''_d \leq c_e$, $e = 1, 2, \ldots, E$, and that $x''_d \in \{0, 1, \ldots, t\}$, $d = 1, 2, \ldots, D$, and, which is contradictive, that $\boldsymbol{y}'' \succ \boldsymbol{y}^k \succeq \boldsymbol{y}^t$. On the other hand suppose that $y'_p < y^k_p = t$. Then $\sum_d f(t, x'_d) > \sum_d f(t, x^k_d)$. However, we also have that $\boldsymbol{y}' \succeq \boldsymbol{y}^t$ (as $t < k - 1$), which implies that $\boldsymbol{y}' = \boldsymbol{y}^t$ or $\boldsymbol{y}' \succ \boldsymbol{y}^t$. If $\boldsymbol{y}' = \boldsymbol{y}^t$ then clearly $\sum_d f(t, x'_d) = \sum_d f(t, x^t_d) = \sum_d (t, x^k_d)$, which is a contradiction. If $\boldsymbol{y}' \succ \boldsymbol{y}^t$ then there must exist an entry s, such that $y'_s > y^t_s$ and $y'_i = y^t_i$, if $i < s$. Now $y^t_i = y^{k-1}_i$ for $i = 1, 2, \ldots, e$, so necessarily $s > p$. But this yields that $\sum_d f(t, x'_d) = \sum_d f(t, x^t_d) = \sum_d f(t, x^k_d)$, which is a contradiction. Hence, if $\boldsymbol{y}^k \succ \boldsymbol{y}'$ there must exist an entry q such that $y^k_q = k$, and $y'_q = k-1$. This in turn implies that $\sum_d f(k, x^k_d) < \sum_d f(k, x'_d) = \lceil \tau_k \rceil$. But $\sum_d f(i, x^k_d) = \sum_d f(i, x'_d) = \lceil \tau_i \rceil$, $1 \leq i \leq k - 1$, as $y^k_i = y'_i$ for $i \leq m$. Hence τ_k cannot be the optimal value of the objective function for P_k, and summarizing, it cannot hold that $\boldsymbol{y}^k \succ \boldsymbol{y}'$. Thus $\boldsymbol{y}' \succeq \boldsymbol{y}^k$. By induction over k, it follows that $\boldsymbol{y}' = \boldsymbol{y}^r$. □

It is an immediate consequence that if we put as an explicit constraint in P_k, $k = 1, 2, \ldots, r$, that $x_d \in \mathbb{Z}^+$, $d = 1, 2, \ldots, D$, then the algorithm is guaranteed to solve problem (7)-(9) with $i = r$. Thus this is an option for an instance for which Algorithm 1 does not produce an integral solution. However, this makes P_k a Mixed Integer Programming (MIP) problem.

There are some implementational issues of Algorithm 1 that ought to be mentioned. First of all, it is convenient to recycle the sparse constraint matrices of the successive LPs, as they change only marginally between consecutive steps. Secondly, special care should be taken in the rounding of τ. For large instances (many variables), aggregation of small numerical errors in the computed variables may cause an erroneous rounding of τ (typically making the right-hand side of (4) too large). Finally, it is essential that the LPs are solved for vertex solutions, as is done by Simplex. This is easily realized if one considers a network of two vertices connected by one edge of capacity 1. Assume that there are two demands between the vertices. As opposed to Simplex, an interior point solution to this instance cannot belong to $\{0, 1\}^2$.

5 A Numerical Experiment

In this section we apply Algorithm 1 (the original LP-based version) to randomly generated problem instances ranging from 36 to 435 demands (corresponding to demands between all vertex-pairs in a 9-vertex network to all vertex-pairs in a 30-vertex network). In all instances $r = 50$ and each edge capacity, c_e, $e = 1, 2, \ldots, E$ belongs to $\{5, 10, 15, \ldots, 50\}$. The results can be found in Table 1. The first 4 columns give, in turn, the number of vertices, V, the number of edges, E, the number of demands, D, and the average length (hops) of a path, $\mathbb{E}(|p|)$. The 5:th column indicates if the algorithm halts with an integral solution (int). Column 6 gives the number of solved LP:s (iterations) that did not produce an integral solution (NILP), and the 7:th column gives the number of iterations for which τ was rounded up ($\lceil \tau \rceil$), i.e., when optimal τ was non-integral (due to rounding errors there is no one-to-one correspondence between rounded τ's and non-integral solutions). The following two columns give the total running time (t) and the required number of iterations (it), respectively. The columns are then repeated for more instances. The computations were carried out on a PC with an Intel PIII-1GHz CPU, RAM of 256 MB, and Windows 2000 OS. The algorithm was implemented in MATLAB6.5, and the LPs are solved using a MATLAB interface (mex-function) to CPLEX 9 (Simplex LP-solver).

Although Algorithm 1 performs satisfactorily on all of the instances considered in Table 1, there exist situations for which it fails. Consider for instance the network given in Figure 3(c). Suppose that the same demands and paths as in Example 4 are given, and that edge capacities are all equal to 1. Then the (sorted) solution generated by Algorithm 1 is $\boldsymbol{y}' = (0.5, 0.5, 0.5)$ but the true OIS is $\boldsymbol{y}^z = (0, 0, 1)$.

Table 1. Testing the algorithm

V	E	D	$\mathbb{E}(\|p\|)$	int	NILP	$\lceil\tau\rceil$	$t(s)$	it	V	E	D	$\mathbb{E}(\|p\|)$	int	NILP	$\lceil\tau\rceil$	$t(s)$	it
9	20	36	5.3	yes	0	0	2.50	26	20	41	190	10.8	yes	0	0	14.95	28
10	20	45	4.7	yes	0	0	7.03	43	21	50	210	11.8	yes	1	1	47.22	42
11	21	55	6.4	yes	1	1	9.47	46	22	55	231	11.7	yes	0	1	40.28	36
12	27	66	6.8	yes	4	1	11.56	41	23	59	253	13.6	yes	1	14	53.77	40
13	28	78	7.2	yes	0	7	14.16	48	24	54	279	13.7	yes	2	10	55.41	37
14	34	91	7.7	yes	1	1	10.27	35	25	54	300	13.7	yes	0	0	43.43	32
15	29	105	9.0	yes	0	0	11.29	37	26	72	325	14.4	yes	0	0	147.34	50
16	34	120	8.6	yes	2	2	17.42	38	27	66	351	14.6	yes	0	0	38.77	26
17	39	136	8.8	yes	0	0	16.86	34	28	69	378	15.7	yes	0	0	144.45	42
18	43	153	10.3	yes	2	7	16.59	30	29	59	406	7.3	yes	0	0	351.72	49
19	45	171	10.3	yes	0	0	18.74	36	30	65	435	8.5	yes	1	0	426.30	49

6 Conclusions

This paper presents a study that addresses a realistic modification of the classical max-min fairness bandwidth assignment problem. As in the classical problem, we consider a set of node-pairs (demands) that are to be assigned bandwidth on fixed single paths in a capacitated network, such that the resulting distribution of flows among node-pairs is max-min fair. However, in this paper it is additionaly assumed that a demand can only be assigned a modular flow volume. The solution to the modified problem is first characterized. Then it is shown that even though the classical problem is solvable in polynomial time, the modified problem is $\mathcal{NP}$-hard. An algorithm based on linear programming (necessarily Simplex) is described and shown to be useful, both in terms of solution quality and running times, for a number of example instances. We prove that if this algorithm (in its basic linear programming form) produces an integral solution, then this must be the solution to the considered problem. It follows that the algorithm can, of course at the cost of increasing comlexity, instead be based on mixed integer programming, and then guarantee that the output is optimal.

References

1. Bertsekas, D., Gallager, R.: Data Networks. Prentice Hall (1987)
2. Nace, D.: A linear programming based approach for computing optimal fair splittable routing. In: IEEE International Symposium on Computers and Communications. (2002) 468474
3. Pióro, M., Nilsson, P., Kubilinskas, E., Fodor, G.: On efficient max-min fair routing algorithms. In: IEEE International Symposium on Computers and Communications. (2003) 465472
4. Pióro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufmann (Elsevier) (2004)
5. Kleinberg, J., Rabani, Y., Tardos, E.: Fairness in routing and load balancing. Journal of Computer and System Sciences 63(1) (2001) 2–20
6. Nilsson, P., Pióro, M.: Unsplittable max-min demand allocation – a routing problem. In: HETNETs05. (2005) P26.
7. Garey, M., Johnson, D.: Computers and intractability – a guide to the theory of NP-completeness. Freeman (1979)
8. Nilsson, P.: Some simple special cases of FIXMMF-MF. Technical report, Dept. of Communication Sytems, Lund University (2005)
9. Ogryczak, W., Pioro, M., Tomaszewski, A.: Telecommunications network design and max-min optimization problem. Journal of telecommunications and information technology 3 (2005)

Anticipatory Distributed Packet Filter Configuration for Carrier-Grade IP-Networks*

Birger Toedtmann and Erwin P. Rathgeb

Computer Networking Technology Group,
Institute of Experimental Mathematics,
Duisburg-Essen University, Germany
{btoedtmann, erwin.rathgeb}@iem.uni-due.de

Abstract. Packet filters have traditionally been used to shield IP networks from known attack flows, ususally within firewall systems connecting trusted and non-trusted network segments. As IP networks grow and tend to connect to more and more neighbor networks with unknown trust status, carrier-grade operators in particular are beginning to experience raising costs due to increasingly complex filter configurations that have to be applied to their networks, in order to maintain a desired security level. In this paper, we present a discussion on the general properties of distributed packet filter configurations and an algorithm for a simplified compilation of anticipatory static packet filter configurations in heterogenous IP networks.

Keywords: Network Security, IP Spoofing, Packet Filters, Critical Infrastructure Protection.

1 Introduction

Over the past years, operators of private and public IP networks have seen an increased amount of security related incidents, ranging from the rare targeted break-in attempt to the more frequent worm and virus spread. One method to protect against these threats is to set up and maintain special traffic-examination and -blocking functions at the edges of the network. The more sophisticated class of systems providing such functions are commonly called 'firewalls', which are often not only capable of simple packet-by-packet filtering but can also handle the inspection of the content of an entire connection.

The major benefit in deploying firewalls is an organizational one: maintain one system that keeps out unwanted traffic (and the malicious content it would import otherwise) instead of individually securing hundreds or even thousands of end-systems inside the network. However, this is only reasonable in an economic sense if the border between trusted parts of the network and non-trusted parts is known and if the number of links to from one part to the other is comparatively

* This work was partially funded by the Bundesministerium für Bildung und Forschung of the Federal Republic of Germany under contract 01AK045. The authors alone are responsible for the content of the paper.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 928–941, 2006.

small. Large carrier-grade IP network operators in particular are confronted with the problem that they have many interconnection points to other networks and must also support a very high traffic throughput at these points. This makes setting up and maintaining firewall systems at interconnection points a prohibitively costly task. Furthermore, borders are not as static any more as in the past, because when network operators grow and merge, the borders of their networks move. Nevertheless, IP carriers have an increased demand for filter functions especially to shield internal management communication driving their networks from being disrupted by denial-of-service (DoS) attacks [1, 2]. To meet this demand without having to deploy a set of expensive firewalls, operators usually fall back on the capabilities of commercial off-the-shelf routing and switching platforms to filter packets. This is often done in a very simple way by configuring filter rules on interfaces line by line within the routers or switches command line interface. A drawback of this method is that it is difficult to automate, especially in heterogenous, multi-vendor environments where filter configurations often have to be adapted to meet the routing platforms specific configuration syntax: as packet filter configuration has never been standardized in IETF management working groups, many operators still maintain packet filter rule sets semi-automatically or even manually.

As a consequence, the need for a flexible mechanism that computes effective filter points (nodes and interfaces) and provides syntactically correct filter statements for the platforms within these network is growing. Our contribution in this paper is an investigation of a new method that automatically finds efficient filter placements for large, carrier-grade, IP networks with heterogenous components. We reconcile the filter-based protection against potential attack flows with anticipated network behaviour upon failure states, where independent routing plans provide resilience. Our method allows for arbitrary threat and use scenarios for a given network and incorporates the diverse, varying filtering capabilites of the nodes inside the network as well as the syntax needed to configure filtering behavior in nodes.

1.1 Related Work

In recent years, there has been a fair amount of research on packet filter configuration issues and firewall technology, however, these approaches most commonly focus either on platform/technology specific problems (developments from firewall vendors) or investigate issues that arise after filter rule sets have been applied. In particular, policy management has been researched, e.g. conflicts that may result from distributed rule sets and how to resolve them [8, 5]. Although the distribution of packet filters in networks has been suggested earlier [6, 9], it was, however, without incorporating the topologic effects that we investigate and describe in this paper. Automatic packet filter compilation for firewall systems has been researched [6, 7], but also without considering topologic effects.

The current state in the area of automated packet filter configuration in multi-vendor environments is that there exists no Management Information Base (MIB) that allows setting filter rules via the Simple Network Management

Protocol (SNMP).[1] The Common Open Policy Service (COPS), which has been developed for policy-based networking and supports the configuration of classification statements, lacks a method for defining security related actions that are not IPsec-specific [10]. Middlebox Communications (MIDCOM), Simple Middlebox Configuration (SIMCO) and NAT/Firewall NSIS Signaling Layer Protocol (NSLP) are newer standardization efforts that aim for automatic configuration of firewall functions in so-called middleboxes (usually application layer gateways) but are quite heavyweight when it comes to implementation and scaling issues [11, 12, 13, 14]. It is thus still the best way to use the routing or switching platforms command line interface (CLI) when configuring packet filter setups.

2 Model Assumptions and Definitions

The development of packet filters has since the beginning seen many differing approaches and naming conventions. Common ground can be found on the general notion that packet filters are a combination of two functions: a *classifier* and an *action* associated with it. The classifier tries to match a packet to a predefined pattern. Usually only the packet header is inspected for this operation to ensure timely decision making. Classsifiers that additionally check a backlog or history table of connections and packets seen at a previous time are called *stateful.* If the packet header matches the specified pattern, the action assigned to it is invoked upon the packet. In our approach, only blocking actions are used, these are specified with *allow*, *permit* or *accept*, and *disallow*, *deny* or *drop.*

A classifier/action pair is usually denoted as a *filter rule*, whereas a list of such rules is known to be a *rule set.* Some vendors also refer to those rule sets as *access control lists.*[2] When a packet does not match any of the individual rules in such an (ordered) list, a default rule, also known as *filter policy* applies. A policy that implicitly drops all non-matching traffic is called *whitelisting,* whereas a policy that accepts all non-matching traffic is known as *blacklisting.* Within the remainder of this paper, we will use simple (non-stateful) disallowing filter rules and blacklisting as we only state explicit prohibits on packets that match a specific pattern. In the following, we investigate rule sets that are distributed over a subset of nodes comprising an IP network, thereby assembling a *distributed packet filter configuration* that enforces a specific global filter policy with local packet filters without requiring a deployment of singular firewall systems.

2.1 Direction-Based Filtering

Adversaries have long since adapted to the existence of packet filter systems and thus have developed their own set of techniques to circumvent them as far as possible. One method is to let the injected packets just look like legitimate packets – this is possible because IP networks allow every user to craft the packets

[1] The RMON MIB does provide the filter group, however the only possible action specified for RMON is capturing the packets that match a filter pattern.

[2] E.g. Cisco, Juniper Networks.

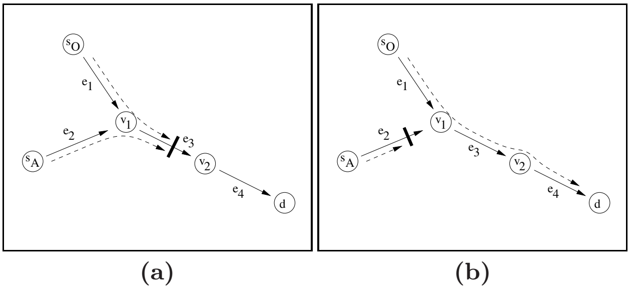


Fig. 1. Direction based filter placement. In (a), packets sent from operator node s_O and attacking node s_A that match the prohibitive filter on edge e_3 will both be discarded, resulting in a lost connection from s_O to d, but also preventing a succesful attack. In (b), the filter has been moved onto edge e_2 where it will only discard attack packets, and not legitimate user packets.

they are going to send into the network themselves. This technique has long been known as "spoofing" [15] and is still quite popular despite increased deployment of anti-spoofing mechanisms in modern access networks; mostly because these types of networks still account only for a small fraction of all vulnerable hosts within the Internet [3, 4]. As a consequence, when activating allowing and disallowing filters on an interface of a network node, the operator faces a trade-off concerning legitimate and malicious traffic that traverses this interface: if accepting filters are active, malicios data packets crafted by the adversary to match the configured pattern within the classifier will be falsely allowed into the network. We call this a *false negative* filter decision. On the other hand, prohibitive filters that have been placed on a path where legitimate packets travel will discard them, usually terminating a favored connection. This case we call a *false positive* filter decision. An operator therefore must anticipate the directions where the legitmate and malicious packet flows will most likely come from, to minimize the costs incurred by either false positives or false negatives. This is illustrated by Fig. 1 where the alternative filter placements influence future potential damages for the operator.

The underlying problem is thus to find the minimum costs associated with each packet filter configuration in terms of this trade-off. Formally speaking, we have

- source nodes s_O(operator) and s_A(adversary) and destination node d
- paths $p_O \in P_O$, which is the path set for all paths from s_O to d
- paths $p_A \in P_A$, which is the path set for all paths from s_A to d
- probability ω_{p_o} of a false positive case that a filter wrongly terminates a connection, this is a compound of the initial probability that this connection itself is up and that it is filtered somewhere on the path p_O
- damage D_O that is incurred if this connection to a service, e.g. SSH from management system to managed control node, is lost due to a misplaced filter
- operational risk $R_O = \omega_{p_o} D_O$

- probability φ_{p_A} of a false negative case where an attacker succeeds in sending packets to the destination node, this is a compound of the initial probability that the attacker sends packets and that he is not filtered anywhere on the path p_O
- damage D_A that is incurred by disruptions caused by an adversary on needed services, e.g. an overloaded SSH port within a control node due to a missing filter
- attack risk $R_A = \omega_{p_A} D_A$

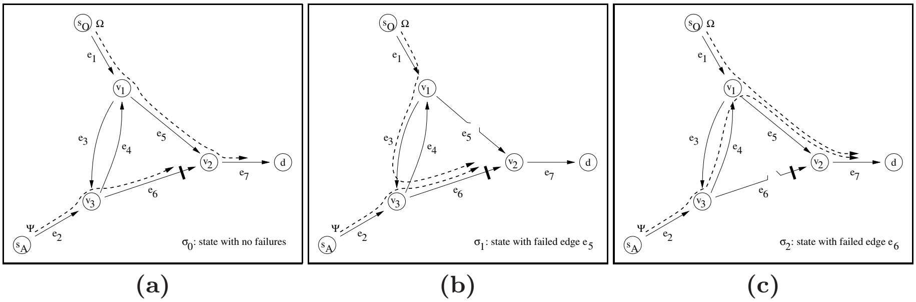
As the costs in terms of the above risks are disjoint for all possible filter configurations, because either a prohibitive filter has been placed there (in this case φ will always be 0) or an allowance filter has been placed (here, ω will always be 0), total costs are additive over the set of valid paths from s_O to d and s_A to d:

$$R_{total} = \sum \omega_{p_o} D_O + \omega_{p_A} D_A, \forall p \in P_O \cap P_A$$

The challenge now is to minimize the total risk for the operator by choosing an efficient distributed packet filter configuration.

2.2 Routing Interference

Network operators are usually more concerned with availability issues than with security issues; however, when it comes to distributed packet filter configurations, both requirements overlap significantly. As we have established in the preceeding section, the major task when trying to minimize false negatives and false positives is to reliably determine sources of legitimate and malicious traffic flows. Unfortunately, in carrier-grade networks these sources change quite frequently as network components fail and resilience mechanisms set up alternative paths. As a consequence, the probability of a specific traffic flow to appear at a specific network nodes interface also depends upon the failure probabilities of the network components and the characteristics of the resilience mechanisms in place. In IP networks, the most important resilience mechanism is its destination-based, hop-by-hop routing mechanism. It determines, based on a routing plan, within all forwarding nodes the best next hop for known destination networks. When a network component – a node or a link – fails, a routing algorithm adjusts to the new network state and disseminates the information about new best next hops, which are then stored in the forwarding table. The difficulty with this resilience mechanism and static packet filter configurations is illustrated in Fig. 2: when a failure occurs, packet flows may be directed over alternative paths, which may result in the wrong flows being dropped (giving a false positive) or accepted (giving a false negative). When trying to integrate a distributed filter configuration with a resilience method based on routing, one must take these trade-offs into account. The major problem is thus to incorporate the routing plan for as many network states as possible to get the corresponding path sets.



s_O Ω
e_1
v_1
e_3
e_4
e_5
v_2
e_7
d
v_3
e_6
Ψ
s_A
e_2
σ_0: state with no failures
σ_1: state with failed edge e_5
σ_2: state with failed edge e_6
(a)
(b)
(c)

Fig. 2. Filter placement and routing interference. In this small example scenario, packets a routed in an destination-based, hop-by-hop fashion. The filter placement on edge e_6 prohibits attack packets from reaching destination node d in the failure-free situation (a), but this changes significantly when edges e_5 or e_6 fail. In (b), legitimate traffic is shifted by the routing plan onto a new path that runs over the filtering edge e_6, resulting in a lost management connection. In (c), attack traffic is wrongy detoured along a non-filtered path, allowing attack packets to reach d.

3 Distributed Packet Filter Computation

Generally speaking, we are in search for a packet filter configuration that protects our network from malicious packet flows coming from a specific attack source. Usually, this attack source is somewhere *outside* our network, whereas the valid packet source (e.g. a management station) is somewhere *inside*. The task is therefore to find a border between outside and inside, and to find one that is *short*, to avoid placing too many filter rules and to keep the number of nodes that enforce the filters small. We are furthermore interested in a flexible mechanism that can cope with shifting security requirements such as changed damage factors or threat levels, i.e. the inital attack probability. Operators in the past simply put filters on their border routers, which is easy (there is no need to specify attack sources, probabilities and damage factors) but in many cases not efficient. The mechanism we describe here is therefore designed to compute a corresponding virtual border by minimizing the total risk as described in section 2.1 and simultaneously keeping the number of filter configurations to deploy as small as possible.

3.1 A Flexible Packet Filter Placement Algorithm

Any approach providing a way to compute direction-based packet filter configurations must incorporate a legitimate, desired usage scenario and a malicious, non-desired threat scenario in order to find a suitable border and place permits and prohibits accordingly. Each of those scenarios will be composed of a flow description and a topologic source specification which indicates from which direction a specific flow is expected to come. In our approach, both use and threat scenario correspond to the same flow description f but provide separate topologic flow source descriptions s_O and s_A. This means that the flow specfcation

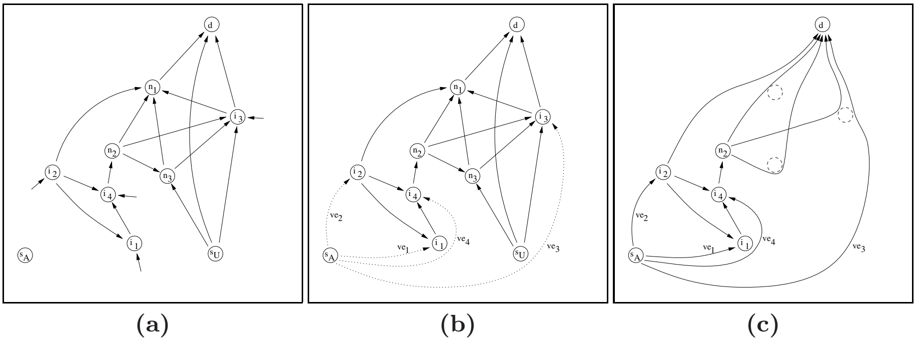


Fig. 3. In (a), we reduce an exemplary network topology to a directed graph with destination node d as the sink and edge routers - the interconnection points i_1- i_4 - together with outside attacking node s_A. In (b), the attacking node is connected to the graph at the interconnection points via virtual edges ve_1- ve_4. When the subgraph containing the dominant operational risk edges has been removed, the residual filter candidate graph emerges as shown in (c).

itself will provide the necessary information for the filter classfiers (IP source and destination addresses, transport protocol, source and destination ports) but it acknowledges that adversaries may craft attack packets that will look exactly like legitimate user packets. In constrast to this, the network specification within the use and threat scenarios will give us the needed differentiation between attack packet streams and user packet streams. This is illustrated by Fig. 3 where the valid source node is known and the operator additionally gives entry points for potential attackers to the network – usually these coincide with the interconnection nodes towards neighbor networks.

As has been outlined in section 2.2, anticipating all paths P_O, P_A that the valid traffic and the attack traffic may take through the network is a requirement for computing where filter placements would be reasonable to reconcile resilience requirements (routing) with security requirements (filtering). Unfortunately, this generally requires a complete network state enumeration for any combination of failed elements, which is of P#-complexity. However, state space reduction is possible if the number of components per path is not too small and the availability of the components is comparatively high [16], which is a very typical characterstic of carrier-grade IP networks. Thus, in our algorithm, we first determine the number of concurrently failed elements we need to inspect, in order to reach a significantly high share of the state space. We then enumerate over all the remaining states and invoke the routing algorithm used for the network for each state, in conjunction with the legitimate use endpoints (s_O, d) and the attack endpoints (s_A, d). We thus yield all two sets of most probable paths for both sources, together with the probability by which they will be effective – this is done by combining the initial flow probability and the availability data for all components respectively. In the next step, we iterate over all paths and over all edges within the paths and add the specific probability of the selected

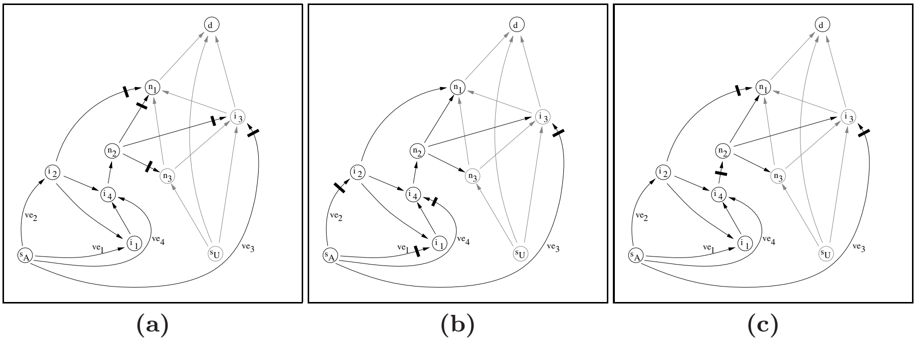


Fig. 4. Filter placement strategies. Backward placement in (a) yields 5 filters near the dominant operational risk set of edges, while the traditional, forward placement (b) needs 4 filters at the networks borders. (c) yields the minimum number of 3 filters by exploiting a focal point inside the network for a more efficient filter placement.

path to the individual edge. As a result, we get a set of edges that additionally contain their legitimate use and attack probability values. Now, we are able to compute the set of *dominant operational risk edges* by comparing the individual operating risk and the attack risk of each edge – assuming that a path will always be filtered on one edge only. This edge set contains all edges where filters should never be active because the risk of wrongly terminating an important management connection is just too high, compared to the acompanying attack risk. All remaining edges of the network comprise the *residual filter candidate graph*: at any edge within this graph, a filter may be placed to prevent adversaries from injecting malicious packets, comprising a virtual border. This is illustrated by Fig. 3 (c), where all dominant operational risk edges have been removed.

Until here, we have reasoned how to assess where it is advisable to place filters and where the costs in terms of operational risk prohibits the placement of filters. In the last step, the actual filter placements are computed. Two ways to find filter placements on the remaining graph are quite obvious: starting from the destination node, going backward over the edges for each path, placing filters as near to the part of the network over which legitime, non-filtered paths run. This approach, illustrated in Fig. 4 (a), reduces the availability of the operators connection to a minimum and provides no direct benefit. The opposite way is to place the filters as near to the attacker as possible, which is the traditional way, moving filter sets to the border of a network as depicted in 4 (b). Inspecting Fig. 4 (c), we can see that it is possible to prevent adversaries from injecting packets into our network by placing fewer filters than the traditional border-placement strategy would suggest. Thus one optimization is to compute a *minimal cutting path edge set*. Another variant of this strategy is not to minimize the total number of filters to be set up but to minimize the number of nodes where filters must be configured – which is the result of a *minimal cutting path node set*. It is easy to see in the example network that a minimal cutting path node set is $\{i_2,i_4,i_3\}$ or $\{n_2,n_1,i_3\}$. This indicates that we usually will get more alternatives here, raising the opportunity to optimize based on filter costs that can be set by the operator. If operators need to upgrade their routing platforms in order to deploy extensive packet filter setups, they may prefer to keep the number of upgrades small and they may prefer to choose the least costly upgrades: if upgrades for the nodes $n_1$and n_2 are cheaper than for the interconnection points, they will prefer the latter variant. Algorithm 1. represents a method to compute an efficient virtual border for filter placements based on the minimal filter number variant.

Algorithm 1. Filter placement by creating a short virtual border

```
(Step 0: Extract state space $\Theta$)
(Step 1: Extract path sets)
for all sources $s_O \in S_O, s_A \in S_A$ do
  for all states $\sigma \in \Theta$ do
    $P_O \leftarrow R_\sigma(s_O, d)$
    $P_A \leftarrow R_\sigma(s_A, d)$
(Step 2: Compute edge utilization)
for all paths $p \in P_O \cap P_A$ do
  for all edges $\epsilon \in p$ do
    if $p \in P_O$ then
      add $availability(p)$ to $\omega_\epsilon$
    if $\epsilon \in P_A$ then
      add $availability(p)$ to $\varphi_\epsilon$
(Step 3: Compute risk distance)
for all paths $p \in P_A$ do
  for all edges $\epsilon \in p$ do
    if $\omega_\epsilon D_O < \varphi_\epsilon D_A$ then
      filter candidate set $F_c \leftarrow (\epsilon, c_\epsilon = 0)$
      for all paths $p \in P_A$ do
        if $\epsilon \in p$ then
          add 1 to edge candidate count $c_\epsilon$
        choose $\epsilon$ from $F_c$ with highest $c_\epsilon$ (it is cutting the most paths)
        for all paths $p \in P_A$ do
          if $\epsilon \in p$ then
            $P_A = P_A - p$
if $P_A \neq \emptyset$ then
  goto Step 3
```

The complexity of this algorithm can be influenced quite heavily by the operator. Shortest path computations exhibit $O(m \times log(n))$ each for adverse topologies [19] and have to run over the selected state space which can amount to $O(2^n)$ if fully explored. The computations of the risk distances are of linear complexity and a formally correct implemented minimal cut on the residual path set will run $O(\sqrt{n} \times m)$ [20], but both have to be invoked only once per f and are thus neglibible. If the state space is confined to a sensible proportion with respect to the discriminating risk function, the overall computational time for each f considered usually is within an acceptable timeframe – for the example depicted by Fig. 6, computations were well below 10 seconds for 5% of the state space. However, path exploration on the state space remains a critical issue for the method used here.[3]

[3] A more exhaustive discussion on algorithmic complexity and running times for test topologies is beyond the scope and limited space of this paper.

3.2 Incorporating Platform Capabilities

Until now we have investigated how a distributed packet filter configuration that locally enforces a global filter policy can be compiled in an efficient way. However, operators often face the problem that platforms do not share the same capabilities, which are subject to the installed software release or acquired feature set. As a consequence, networks may include nodes that are not capable of filtering the considered packet flow f, for one of the following reasons:

- the node is technically not able to classify for f
- the node is technically not able to execute a drop or accept action on packets that match f
- the node is principally capable of filtering packets that match f, however, crucial information needed by the classifier is not available (e.g. IP addresses describing f cannot be obtained)
- the operator does not want the node to filter packets matching f, because of performance considerations

Algorithm 2. Filter placement with capability integration

```
(Step 3: Compute risk distance)
...
if $\omega_\epsilon D_O < \varphi_\epsilon D_A$ and $\epsilon \in C$ (the set of filter capable edges) then
    $\Gamma_A = \Gamma_A - \epsilon$
    ...
if $P_A \neq \emptyset$ and $\Gamma_A \neq \emptyset$ then
    goto step 3
if $P_A \neq \emptyset$ then
    Issue notice: "remaining attack paths $P_A$ not filtered!"
```

We can easily incorporate this case into our existing framework by first checking all edges in the network for their respective filter capabilities, excluding those edges that do not have their source in a node that can put outgoing filters on their respective interface and the destination in node that cannot put ingoing filters on the respective interface. The resulting filter capable edge set C is used within Algorithm 2. to avoid edges where filters cannot be set up. If the attack path set cannot be emptied after all candidate edges have been investigated, the user will be noticed that an open attack path still exists.

3.3 Prototype Implementation

The algorithm described above was implemented as a component of a larger management process that takes global access policies for packet flows and converts them to local, platform specific filter rule sets for a given network. We developed a Java application called Access Policy Configuration Point (APCP), that reads a network specification including all nodes within the network and their connections to each other as well as platform type and operating system

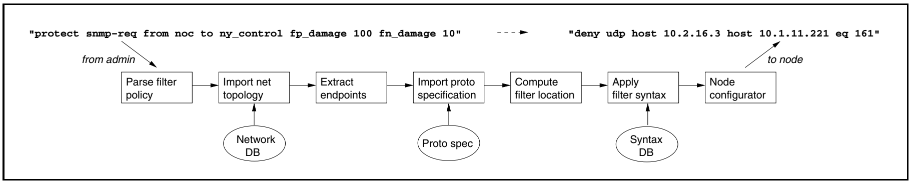


Fig. 5. APCP build process for distributed packet filters. After reading the filter policy statements and the network description as well as the threat specification, the APCP expands all endpoints and the application protocol to build the flow specification and compute the best virtual border. At the end, a syntax database is consulted to yield applicable filter statements for all platforms where filters will be configured.

version – a network discovery process providing this specification for a live network was also a part of the application. The user has to provide policy strings, enhanced by damage factors:

"protect <application protocol handle> from <source node/group handle> to <destination node/group handle> fp_ damage <factor> fn_ damage <factor>".

When these policy specifications have been entered and the network description file, which also included a threat specification (subnet specifications for external, untrusted networks plus attack probability) has been read, the APCP expands the endpoints for the calculation of the virtual border where filter placement would be most effective. An example is illustrated in Fig. 5 where the strings "noc" and "ny_control" have been expanded to 10.2.16.3 and 10.1.11.221.

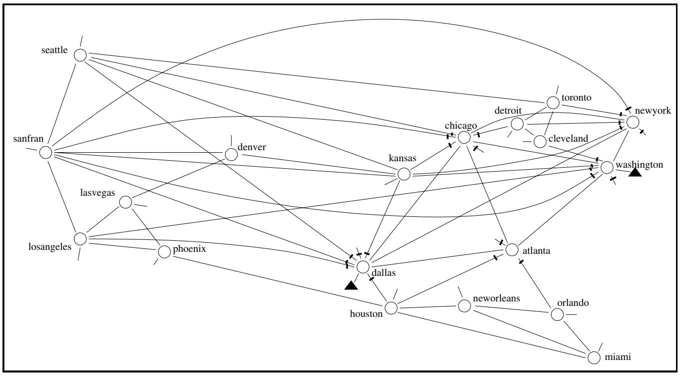


Fig. 6. Distributed filter configuration example. This case relates to a simplified topology from UUNET (1997). In order to protect a connection between a management station at Dallas and a control node at Washington, the APCP set up a new virtual border in the east, reducing the number of filter nodes significantly. The software also warned when focal routers such as Chicago were marked as non-filter capable.

In the next step the protocol handle is expanded to yield a complete flow specification f (in our example udp:10.2.16.3:*:10.1.11.221:161). Afterwards, the algorithm described in section 3 is invoked to compute an efficient filter placement for the network.

The finishing steps are then to convert the information needed for the classfier into the syntactically correct format for each of the target platforms and to export the configuration statements into the nodes themselves. Currently we are able to provide conversion rules for Cisco IOS, Juniper ERX and Linux platforms. As a case study, we took a topology known from UUNET and applied a use and a threat scenario where an internal management connection has to be protected against attack sources placed at the borders, as illustrated by Fig. 6.

3.4 Discussion

Our mechanism allows a flexible computation of efficient packet filter placements along a virtual border within the network, with respect to given usage and threat scenarios and a weighting function (by damage factor). However, one might argue against *anticipatory, static* packet filter configurations we have presented here, favoring *adaptive, dynamic* packet filter mechanisms, because the best method against routing interference is, of course, to incorporate the network state and the accompanying routing decisions into the filter placement decision just-in-time. To give an example following the case depicted in Fig. 2: a drop filter should only be in placed and active in node v_2, if edge e_5 is operational. Particularly when using link-state routing algorithms, which require every node to contain a full view of the complete network state, it is possible to create ad-hoc filtering decisions on the routing information. This has also been suggested in a slightly different manner in [17]. However, several reasons can be put forward against such a mechanism:

1. In contrast to routing, operators usually do not want an automated, self-adapting filtering mechanism, simply because of the disruptive effects of drop filters.
2. When a change in the topology of the network occurs, routing algorithms always have a convergence period. Within this period, dynamically placed packet filters may wrongly discard packets (similar to the micro-loop effect seen with OSPF [18]; when using protocols such as BGP, the convergence period is even within minutes).
3. Currently, there exists no single routing or switching platform that implements dynamic, network state dependent packet filters.

Thus our approach is more suitable for the problems carrier-grade operators face currently, even with the input overhead needed in contrast to the traditional, border placement of filters. However, in terms of computational complexity, the runtime behaviour for Step 1 – the extraction of all possible path sets – still needs improvements. When not using an incremental routing algorithm [19], a full convergence for each failure condition set by Step 0 is needed, which is for networks with hundreds of nodes in the range of tens of seconds each.

When this is multiplied by the number of states needed for a significant share of the complete state space, computation time reaches tens of minutes, in adverse circumstances even more than an hour.

4 Conclusions

In this paper we presented a new, flexible mechanism for computing distributed packet filter configurations for large, heterogenous IP networks. Instead of placing filters at outer borders only, our algorithm usually finds a more efficient virtual border, reducing the number of filters needed. We integrated a filter capability detection method in order to maintain a tight filter setup despite nodes not being available for filter configuration. We implemented the mechanism and enhanced it with a syntax conversion to meet platform-specific configuration demands and were thus able to demonstrate its usefulness for carrier-grade, multi-vendor environments.

For future work, we plan to evaluate the suitability of the mechanism for different topology sets and varying usage and threat scenarios. Furthermore, we will expand the set of filter actions, which are not restricted to drop and accept filters only, but can, depending on the purpose of the individual filter setup, include rate limiting, normalizing and cryptographic processing as well.

References

[1] Ferguson, P., Senie, D., "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", IETF Request for Comments (RFC) 2827, 2000

[2] Rescorla, E., Handley, M., "Internet Denial of Service Considerations", Internet Draft, IETF Spetember 2005.

[3] Pang, R., Yegneswaran, V., Barford, P., Paxson, V., Peterson, L. "Characteristics of Internet Background Radiation", Proceedings of the ACM Internet Measurement Conference, October, 2004.

[4] Dietrich, D. "Bogons and Bogon Filtering", Presentation at NANOG33, February 2005, http://www.nanog.org/mtg-0501/pdf/deitrich.pdf.

[5] Al-Shaer, E.S., Hamed, H.H., "Discovery of Policy Anomalies in Distributed Firewalls, INFOCOM 2004.

[6] Guttman, J., "Filtering Posture: Local Enforcement of Global Policies", Proceedings of the 1997 IEEE Symposium on Security and Privacy, May 1997.

[7] Bartal, Y., Mayer, A., Nissim, K., Wool, A., "Firmato: A Novel Firewall Management Toolkit", Proceedings of the 1999 IEEE Symposium on Security and Privacy.

[8] Hinrichs, S., "Policy-Based Management: Bridging the Gap", Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC '99), December 1999.

[9] Ioannidis, S., Keromytis, A., Bellovin, S., Smith, J., "Implementing a Distributed Firewall", Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS '00), November 2000.

[10] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R., Sastry, A.: "The COPS (Common Open Policy Service) Protocol", RFC 2748, IETF January 2000

[11] P. Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., Rayan, A., "Middlebox communication architecture and framework", RFC 3303, IETF August 2002.

[12] Stiemerling, M., Quittek, J., Taylor, T., "Middlebox Communications (MIDCOM) Protocol Semantics", RFC 3989, IETF February 2005.

[13] Stiemerling, M., Quittek, J., "Simple Middlebox Configuration (SIMCO) Protocol Version 3.0", Internet Draft, IETF September 2004.

[14] Stiemerling, M., Tschofenig, H., Aoun, C., "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", Internet Draft, IETF July 2005.

[15] Heberlein, L.T., "Attack Class: Address Spoofing", Proceedings of the Nineteenth National Information Systems Security Conference pp. 371-377, October 1996.

[16] Li, V.O.K., Silvester, J.A., "Performance Analysis of Networks with Unreliable Components", IEEE Transactions on Communications, Vol. 32, No 10, October 1984, pp1105-1110

[17] Park, K., Lee, H., "A Proactive Approach to Distributed DoS Attack Prevention using Route-Based Packet Filtering", Tech. Rep. CSD-00-017, Department of Computer Sciences, Purdue University, December 2000.

[18] Francois, P., Bonaventure, O. "Avoiding transient loops during IGP convergence in IP networks", IEEE INFOCOM 2005.

[19] Narváez, P. , Siu, K.Y., Tzeng, H.Y., "New dynamic algorithms for shortest path tree computation", IEEE/ACM Transactions on Networking (TON), Vol. 8, p.734-746, December 2000.

[20] Even, S., Tarjan, E.R., "Network flow and testing graph connectivity", Siam Journal on Computing, Vol. 4, p507-518, 1975.

Fast Handoff Scheme for Seamless Multimedia Service in Wireless LAN

Hye-Soo Kim, Sang-Hee Park, Chun-Su Park,
Jae-Won Kim, and Sung-Jea Ko

Department of Electronics Engineering, Korea University,
Anam-Dong Sungbuk-Ku, Seoul, Korea
{hyesoo, jerry, cspark, jw9557, sjko}@dali.korea.ac.kr

Abstract. In this paper[1] , we propose a fast handoff method for seamless multimedia service in IEEE 802.11 WLAN. The proposed method uses an improved access point (AP) with additional radio frequency (RF) module, SNIFFER, monitoring the movement of the station (STA). By using the SNIFFER, the proposed method can completely remove the probe delay. Furthermore, we also propose an effective handoff decision method taking into account the quality of service (QoS) in the application layer. The proposed method uses packet loss information and the received signal strength indication (RSSI) for the handoff decision. Experimental results show that the proposed method can improve the performance of the seamless multimedia service by drastically reducing the handoff delay and packet loss.

1 Introduction

The main issue of WLAN is to provide seamless host mobility during the handoff that is the mechanism occurred when an STA moves its association from one AP to another [1]-[8]. During a handoff, no frames should be lost, and the data stream to the video client should be kept as smooth as possible. Hence a fast and efficient handoff scheme is needed for real-time multimedia service to maintain connectivity, minimize data loss and latency while crossing cell boundaries during data transfers [2]-[3].

The handoff procedure in the WLAN can be divided into three distinct logical phases: *probing*, *authentication*, and *reassociation*. In the probing phase, an STA scans for APs by sending *ProbeRequest* messages (Active Scanning) or listening for *Beacon* messages (Passive Scanning). After scanning all channels, an AP is selected by the STA using the RSSI. The STA exchanges IEEE 802.11 authentication messages with the selected AP. Finally, if the AP authenticates the STA, an association moves from an old AP to a new AP by exchanging reassociation messages.

The major problem of handoff procedure in WLAN is the *handoff delay* introduced when an STA is unable to communicate with a certain AP or when

[1] This work is supported by a grant from Samsung Advanced Institute of Technology.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 942–953, 2006.

the user mobility increases in the WLAN. The handoff delay in the WLAN is divided into three types; probe delay, authentication delay, and reassociation delay. It is well known that the probe delay is the primary contributor to the overall handoff latency [4]. Thus, the probe delay has to be significantly reduced for the seamless multimedia communications services.

In this paper, we propose a fast handoff method for seamless multimedia service in IEEE 802.11 WLAN. The proposed method uses a modified AP with SNIFFER which is the additional RF module monitoring the movement of the STA. In the conventional handoff method, an STA has to scan all channels to detect the most probable AP in the probing phase. On the other hand, the proposed method can completely remove the probing phase, i.e. *zero probing delay*, since the STA can obtain the information about the adjacent AP by using the SNIFFER module. Moreover, the AP with dual RF module can provide the network information to the STA by using the modified neighbor graph (NG) which represents the set of potential next APs.

We also propose an effective handoff decision method taking into account the QoS in the application layer. In general, the handoff occurs when the STA can not detect enough signal strength from the AP. This handoff method, however, may be inefficient since serious packet losses are often generated even when the measured signal is strong enough. In the proposed handoff method, we use packet loss information for the handoff initiation and the variance of the RSSI for the handoff decision.

This paper is organized as follows. In Section 2, we briefly review the IEEE 802.11, conventional handoff procedure, and NG. In Section 3, the proposed fast handoff method is introduced in detail. The experimental results are shown in Section 4 and we give some concluding remarks in Section 5.

2 Backgrounds

Before introducing the proposed methods, we briefly review the IEEE 802.11 and the concept of the NG, which are the base of the proposed methods.

2.1 IEEE 802.11 Overview

The IEEE 802.11 standard uses a basic service set (BSS) for the basic building block of networks. An STA is free to move within the BSS, but it can no longer communicate directly with other stations if it leaves the BSS. An independent BSS (IBSS) is a standalone BSS that has no backbone infrastructure and consists of at least two wireless stations as shown in Fig. 1 (a). This type of network is often referred to as ad-hoc network because it can be constructed quickly without much planning. The ad-hoc wireless network will satisfy most needs of users occupying a smaller area, such as a single room, sales floor, or hospital wing.

For requirements exceeding the range limitations of the independent BSS, the 802.11 standard defines an extended service set (ESS), which is identified by its service set identifier (SSID), as illustrated in Fig. 1 (b). This type of

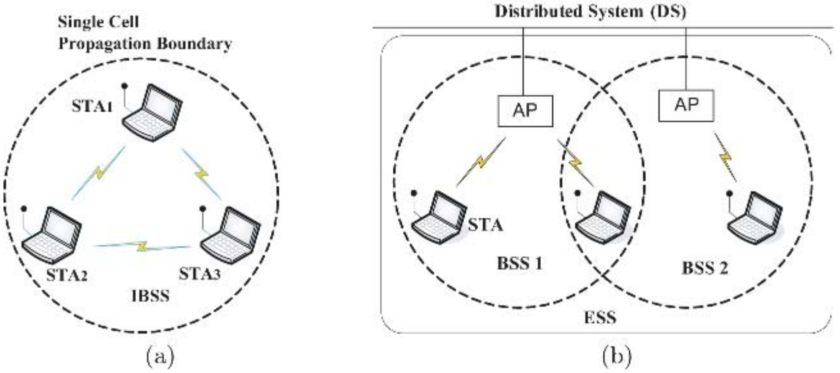


Fig. 1. The IEEE 802.11 topology. (a) Ad-hoc network, (b) Infrastructure network.

configuration satisfies the needs of large coverage networks of arbitrary size and complexity. An ESS of 802.11 WLAN consists of multiple cells interconnected by APs and a distributed system (DS), such as ethernet [1]–[5]. The STA can change the BSS where it is to be connected, using the active/passive scanning and reassociation service. In fact, while an STA is associated with a BSS, it can decide that the connection quality is poor, so it scans the medium to search for a more reliable connection. If the search is successful, it can decide to invoke the reassociation request to a new AP. If the reassociation response is successful, the STA has roamed to the new AP. Normally, an old AP is notified through the DS. Otherwise, if the reassociation request fails, the STA tries to search for a new BSS [6].

2.2 Conventional Handoff Procedure in 802.11 WLAN

A handoff occurs when an STA moves beyond the radio range of one AP, and enters another BSS (at the MAC layer). An STA continuously monitors the signal strength and link quality from the associated AP. If the signal strength is too low, the STA scans all the channels to find a neighboring AP that produces a stronger signal. By switching to another AP referred to as handoff. During the handoff, management frames are exchanged between the STA and the AP. Fig. 2 shows the sequence of messages typically observed during a handoff process. The handoff process starts with the first probe request message and ends with a reassociation response message from an AP.

The complete handoff procedure can be divided into three distinct logical phases: scanning, authentication, and reassociation. During the first phase, an STA scans for APs by either sending *ProbeRequest* messages (Active Scanning) or listening for *Beacon* messages (Passive Scanning). After scanning all the channels, the STA selects an AP using the RSSI, link quality, and etc. The STA exchanges IEEE 802.11 authentication messages with the selected AP. Finally, if the AP authenticates the STA, an association moves from an old AP to a new AP as following steps:

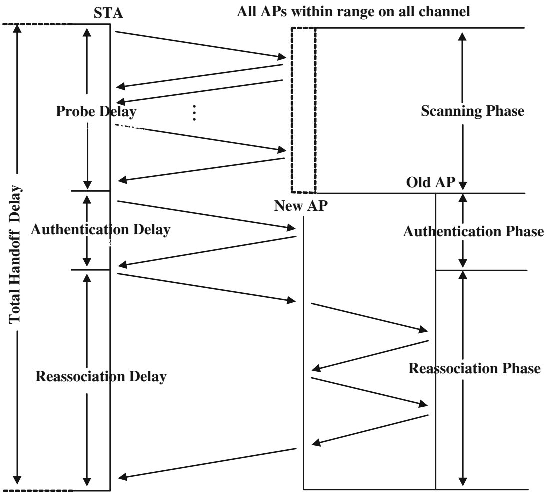


Fig. 2. IEEE 802.11 handoff procedure with IAPP

(1) An STA issues a *ReassociationRequest* message to a new AP. The new AP must communicate with the old AP to confirm that a previous association existed;
(2) The new AP processes the *ReassociationRequest*;
(3) The new AP contacts the old AP to finish the reassociation procedure with IAPP [?];
(4) The old AP sends any buffered frames for the STA to the new AP;
(5) The new AP begins processing frames for the STA.

The delay incurred during these three phases is referred to as the link layer (L2) handoff latency, that consists of probe delay, authentication delay, and reassociation delay. Mishra [4] showed that scanning delay is dominant among three delay. Therefore, to solve the problem of L2 handoff delay, scanning delay has to be reduced.

2.3 Concept of the NG

In this subsection, we describe the notion and motivation for the NG, and the abstractions they provide. Given a wireless network, the NG containing the *reassociation relationship* is constructed. Fig. 3 (a) and (b) show the physical topology of the wireless network and the corresponding NG [7].

Reassociation Relationship: Two APs, ap_i and ap_j, are said to have a *reassociation relationship* if it is possible for an STA to perform an IEEE 802.11 reassociation through some path of motion between the physical locations of ap_i and ap_j. The *reassociation relationship* depends on the placement of APs, signal strength, and other topological factors and in most cases corresponds to the physical distance (vicinity) between the APs.

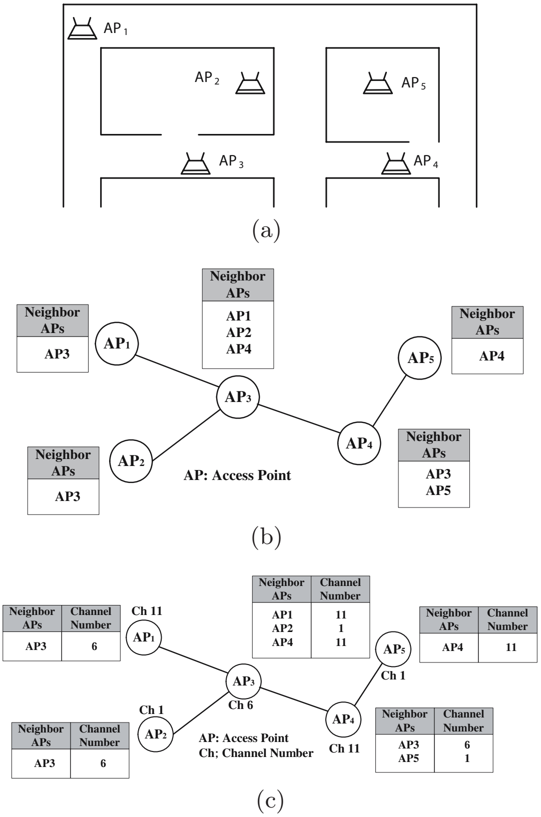


Fig. 3. Concept of the NG. (a) Placement of APs, (b) Conventional NG, (c) Modified NG

Data Structure (NG): Define an undirected graph $G = (V, E)$ where $V = \{ap_1, ap_2, \cdots, ap_n\}$ is the set of all APs constituting the wireless network. And the set E includes all existing edges e_{ij}'s where $e_{ij} = (ap_i; ap_j)$ represents the *reassociation relationship*. There is an edge e_{ij} between ap_i and ap_j if they have a *reassociation relationship*. Define $N(ap_i) = \{ap_{i_k} : ap_{i_k} \in V, e_{ik} \in E\}$, i.e., the set of all neighbors of ap_i in G.

The NG can be implemented either in a centralized or a distributed manner. In this paper, the NG is implemented in a centralized fashion, with correspondent node (CN) storing all the NG data structure. The NG can be automatically generated by the following algorithm with the management message of IEEE 802.11.

(1) If an STA associated with AP_j sends *Reassociate Request* to AP_i, then add an element to both $N(ap_i)$ and $N(ap_j)$ (i.e. an entry in AP_i, for j and vice versa);

(2) If e_{ij} is not included in E, then create new edge. The creation of a new edge requires longer time and can be regard as '*high latency handoff*'. This occurs only once per edge.

The NG proposed in [7] uses the topological information on APs. Our proposed algorithm, however, requires channels of APs as well as topological information. Thus, we modify the data structure of NG as follows:

$$\begin{aligned}
&G' = (V', E), \\
&V' = \{v_i : v_i = (ap_i, channel), v_i \in V\}, \\
&e_{ij} = (ap_i, ap_j), \\
&N(ap_i) = \{ap_{i_k} : ap_{i_k} \in V', e_{ik} \in E\},
\end{aligned} \tag{1}$$

where G' is the modified NG, and V' is the set which consists of APs and their channels. Therefore, we add the channel index to the conventional NG as shown in Fig. 3 (c). In Section 3, we develop a fast handoff algorithm based on the modified NG.

3 Proposed Fast Handoff Method

In this section, we present a fast handoff method with the AP with dual RF modules by using the modified NG described in Section 2.

3.1 Proposed AP with Dual RF Modules

In IEEE 802.11 WLAN, a handoff is controlled by the STA [9]-[10]. Before the handoff to the new AP, an STA continuously monitors the signal strength and link quality of the associated AP. If the signal strength is lower than a certain threshold, the STA scans all channels to find an AP that produces the strongest signal. This procedure is known as the probing (or scanning). It is known that the probe delay is a dominant part of the entire handoff delay.

To remove the probe delay, we propose a new type of handoff method in WLAN. As shown in Fig. 4 (a), the conventional AP contains a single RF module that can receive and transmit signals by turns in the allotted channel.

Since the conventional AP uses only one channel, it can not detect the movement of STA communicating with another AP. For example, as shown in Fig. 4 (c), the AP1 can not detect the STA3 associated with AP2 even when the STA3 enters the BSS of the AP1. Therefore, we add an additional RF module, SNIFFER, to monitor the channels of adjacent APs as shown in Fig. 4 (b). By using the SNIFFER, the proposed AP can eavesdrop the medium access control (MAC) frame of incoming STA3 as shown in Fig. 4 (d). By examining MAC frame received from the STA3, the AP1 can obtain the address of AP2 associated with STA3. Then, the information on AP1 is transferred to the STA3 through AP2. Since the proposed AP can provide the scanning results to the STA, the probe delay at the STA is eliminated.

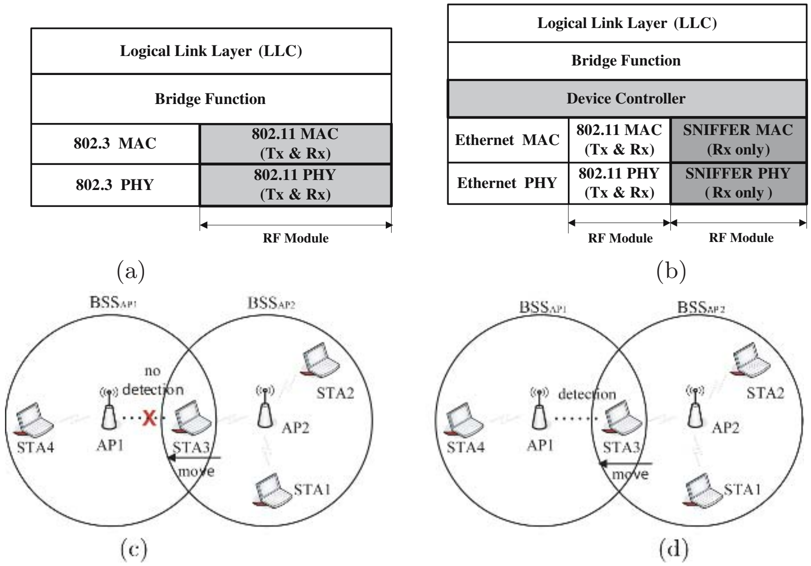


Fig. 4. The architecture of APs. (a) Conventional AP with single RF module, (b) Proposed AP with dual RF modules, (c) Example using conventional AP, (d) Example using proposed AP.

3.2 Proposed Handoff Decision Method

Traditional handoff decision algorithms are based on the RSSI [11]-[12]. If the RSSI is smaller than a certain threshold, the handoff from one AP to the other occurs. However, this handoff decision method has some drawbacks that serious packet losses can be generated even when the RSSI from the associated AP is strong enough. Moreover, a big fluctuation in RSSI causes handoff oscillation, i.e. ping-pong phenomenon, so that the STA wanders between two cells and the quantity of the packet loss increases significantly.

In this subsection, we propose an effective handoff decision method using packet loss ratio (PLR) for the L2 handoff. Fig. 6 shows the relationship between the PLR and RSSI.

As an STA moves away from its current AP, the RSSI measured at the AP decreases, while the PLR increases. In the conventional handoff decision method, the STA does not start the handoff procedure until the RSSI value is lower than a certain threshold. If the threshold is not sufficiently high, serious packet losses can be introduced in the conventional method. On the other hand, the proposed decision method starts the handoff initiation step if the PLR at the STA is greater than a certain threshold. Therefore, the proposed decision method can reduce the quality degradation introduced by the handoff.

In the handoff initiation step, the STA monitors the variations of the RSSI from the current AP and receives that from the SNIFFER of the new AP. Since the RSSI fluctuates over time, we use the median filter with length $2k + 1$ defined as:

$$Z_i(x) = med(R_i(x-k), ..., R_i(x), ..., R_i(x+k)), \tag{2}$$

where x is the position of the STA and $R_i(x)$ is the RSSI value from the AP_i. As the STA becomes closer to the AP2, the median of RSSIs from the AP2 increases, whereas that from the current AP decreases. Since $Z_i(x)$ reflects the direction of the STA and signal strength of the APs, we define the handoff decision parameter, $H(x)$, which is given by

$$H(x) = Z_i(x) - Z_j(x), \tag{3}$$

where $Z_i(x)$ and $Z_j(x)$, respectively, are the median values of RSSI from AP_i and AP_j. If $H(x)$ is smaller than a certain threshold T, the handoff to the new AP is performed. Otherwise, the association with the current AP is maintained.

3.3 Fast Handoff Procedure

The proposed handoff method is performed in both the L2. Fig. 5 illustrates a flow of the proposed handoff method. The AP1 and AP2, respectively, have the communication channels 1 and 6. As the STA moves toward AP2 from AP1, the handoff procedure is performed as follows:

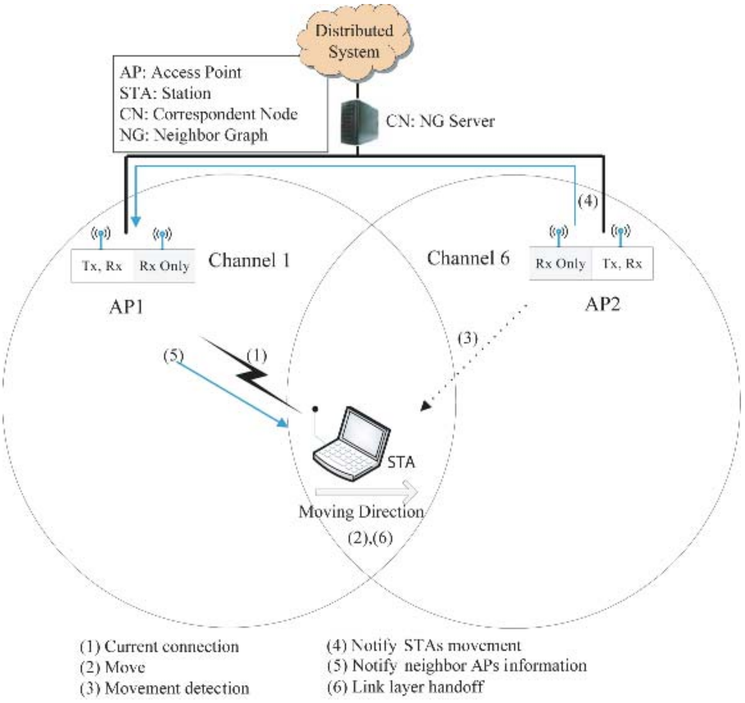


Fig. 5. Procedure of the proposed fast handoff method

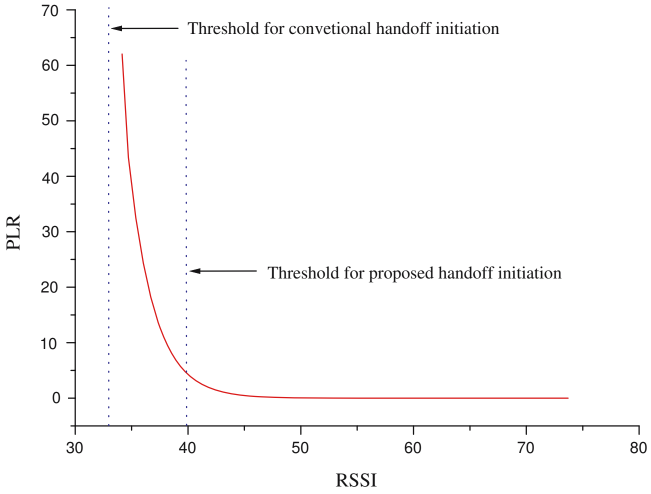


Fig. 6. The relation between the PLR and RSSI

(1), (2) The STA associated with AP1 moves toward AP2;
(3) The SNIFFER module of AP2 receives the MAC frames of the STA;
(4) Using the destination address in the MAC frame of the STA, the SNIFFER of AP2 can be aware of information on the AP1. Then, the information is saved in the NG and transferred to AP1;
(5) By investigating the NG, the AP1 relays the channel information on the AP2 to the STA;
(6) The STA monitors whether packet losses occur or not. If the packet losses are detected, the L2 handoff is initiated. Then, the L2 handoff is performed when $H(x) > T$;

As described in Section 3.1, the proposed handoff algorithm can eliminate the probe delay by using the SNIFFER module added to AP. And the STA can be authenticated and associated before handoff by using the NG. Therefore, the proposed handoff method can drastically reduce the L2 handoff delay.

4 Experimental Results

We developed an experimental platform in order to evaluate the performance of the proposed method. To exchange the NG information, the socket interface is used. The device driver of a common WNIC was modified so that the STA operates as an AP that can support the handoff initiation message. We have developed *NG Server*, *NG Client*, and SNIFFER for the proposed mechanism. The *NG Server* manages the data structure of NG on the experimental platform and processes the request of the *NG Client* that updates the NG information on the STA after the STA moves to the another AP. Using destination address in the MAC frame of the STA, the SNIFFER can be aware of the AP associated with the STA. If the AP can support the requirement of the STA, the SNIFFER sends the old AP a message including handoff information such as the measured RSSI, the available throughput, the MAC address of the STA, and so on.

Fig. 7 shows the median values of RSSIs, $Z_1(x)$ and $Z_2(x)$. As the STA becomes closer to the AP2, $Z_2(x)$ increases, whereas $Z_1(x)$ decreases. To evaluate the performance of the proposed method in terms of handoff delay, we

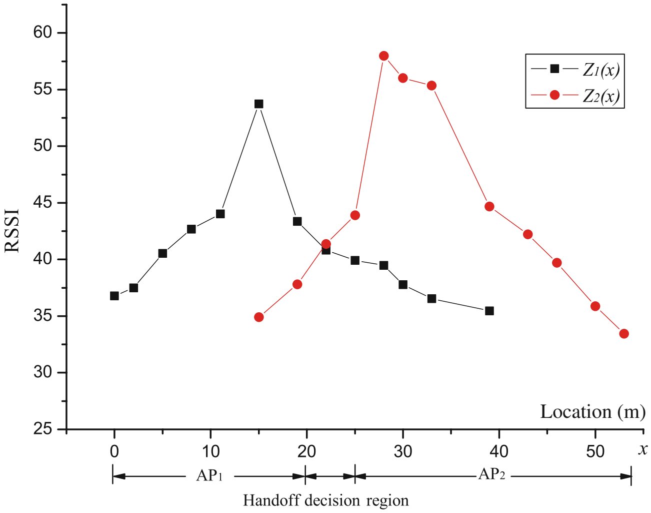


Fig. 7. Median values of RSSI

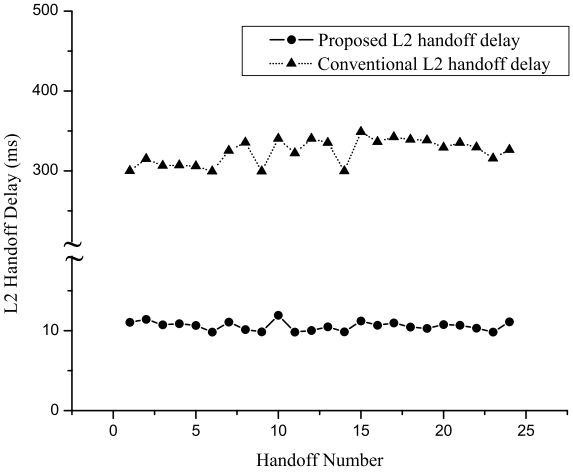


Fig. 8. L2 handoff delay

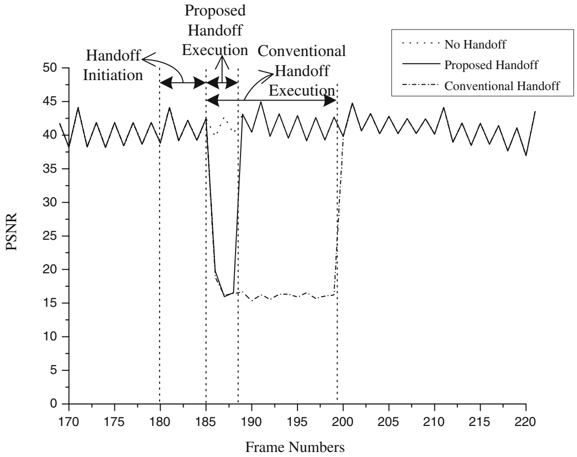


Fig. 9. PSNR during handoff

measured the handoff delay at the L2. As shown in Fig. 8, the value of average L2 handoff delay incurred by the proposed method is much lower than that of the conventional L2 handoff delay [4].

The test sequence of *Akiyo* in QCIF format (176×144) is used to examine the performance of the proposed handoff decision method. The sequence with 300 frames is transmitted to the client at the rate of 30 frames per second. We developed the video streaming server so as to show that the proposed method can improve the QoS during handoff.

Fig. 9 shows the peak signal-to-noise ratio (PSNR) results of the transmitted streaming video when the handoff occurs at the 185 frame. The quality of video in the conventional handoff decision method is significantly degraded just before handoff since there are a lot of packet losses before the handoff initiation. On the other hand, the proposed method can provide seamless video streaming during handoff procedure, since the handoff is initiated before the PLR becomes too low.

5 Conclusions

In this paper, we have presented a fast handoff method for seamless multimedia service using the AP with dual RF modules. Since the SNIFFER module monitors the movement of the STA, the proposed method can remove the probe delay which is the dominant part among the three types of L2 handoff delays. We also have proposed an effective handoff decision method using the PLR and the RSSI. By determining the optimal time to handoff, the proposed method improves the QoS during handoff. Experimental results indicate that seamless multimedia service can be achieved with the proposed method in WLAN.

References

1. Gast, M. S.: 802.11 Wireless Networks. O'REILLY (2002) 1–150
2. ISO/IEC 8802-11 - ANSI/IEEE Std 802.11: Information Technology Part 11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE (1999)
3. Cheng, L. T., Pink, S., Lye, K., M.: A fast handoff scheme for wireless networks. Proceeding of the 2nd ACM international workshop on wireless mobile Multimedia (1999) 83–90
4. Mishra, A., Shin, M., H., Albaugh, W.: An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. ACM SIGCOMM Computer Communication Review **3** (2003) 93–102
5. Geier, J.: Wireless LANs, Second Edition. SAMS **2** (2001)
6. Ramjee, P., Luis, M.: Wireless LANs and WPANs towards 4G wireless. Artech House (2003)
7. Mishra, A., Shin, M., H., Albaugh, W.: Context Caching using Neighbor Graphs for Fast Handoff in a Wireless Network. Computer Science Technical Report CS-TR-4477 (2003)
8. Balachandran, A., Voelke, G., Bahl, P., Rangan, P.: Characterizing User Behavior and Network Performance in a Public Wireless LAN. Proceeding of ACM SIGEMTRICS (2002)
9. Caceres, R., Padmanabhan, V., N.: Fast and Scalable Wireless Handoffs in Support of Mobile Internet Audio. Moblie Networks and Application **3** (1998) 180–188
10. Akyildiz, I., F.: Mobility management in next-generation wireless systems. Proceedings of the IEEE **87** (1999) 1347–1384
11. Chia, S., Warburton, R.: Handoff Creteria for City Microcellular Radio Systems. Proceeding of VTC (1990) 276–281
12. Chia, S.: The control of Handover Initiation in Microcells. Proceeding of VTC (1991) 531–536

On the Tradeoff Between Blocking and Dropping Probabilities in CDMA Networks Supporting Elastic Services

Gábor Fodor[1], Miklós Telek[2], and Leonardo Badia[3]

[1] Ericsson Research, SE-164 80 Stockholm, Sweden
Gabor.Fodor@ericsson.com
[2] Budapest University of Technology and Economics, H-1111 Budapest, Hungary
telek@hit.bme.hu
[3] Consorzio Ferrara Ricerche, 44100 Ferrara, Italy
lbadia@ing.unife.it

Abstract. This paper is a sequel of previous work, in which we proposed a model and computational technique to calculate the Erlang capacity of a single CDMA cell that supports elastic services. The present paper extends that base model by taking into account two important features of CDMA. First, we capture the impact of *soft blocking* by modeling the neighbor cell interference as a lognormally distributed random variable. Secondly, we model the impact of the outage by taking into account that in-progress sessions can be *dropped* with a probability that depends on the current load in the system. We then consider a system with elastic and rigid service classes and analyze the trade-off between the total (soft and hard) blocking probabilities on the one hand and the throughput and the session drop probabilities on the other.

1 Introduction

The teletraffic behavior of code division multiple access (CDMA) networks has been the topic of research ever since CDMA started to gain popularity for military and commercial applications, see for instance Chapter 6 of [1] (and the references therein) that are concerned with the Erlang capacity of CDMA networks. The paper by Evans and Everitt used an $M/G/\infty$ queue model to assess the uplink capacity of CDMA cellular networks and also presented a technique to calculate the outage probability [2]. These classical papers have focused on ”rigid” traffic in the sense that elastic or best effort traffic whose bit rate can dynamically change was not part of the models. Subsequently, the seminal paper by Altman proposed a Shannon like capacity measure called the ”best effort capacity” that explicitly takes into account the behavior of elastic sessions [3].

The importance of modeling outages and *session drops* and their impacts on the Erlang capacity in cellular networks in general and in CDMA in particular has been emphasized by several authors, see for instance [2] and more recently [7]. Session drops are primarily caused by outages, when the desired signal-to-noise ratio for a session stays under a predefined threshold during such a long time that the session gets interrupted. However, sessions can be dropped by a load control algorithm (typically located in the radio network controller in WCDMA) to preserve system stability. Session

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 954–965, 2006.

interruptions are perceived negatively by end users - more negatively than blocking a session - and therefore their probability should be minimized by suitable resource management (including admission control) techniques.

The purpose of this paper is to develop a model that can be used to analyze the trade-off between the blocking and dropping probabilities in CDMA in the presence of elastic traffic. We build on the model developed for elastic traffic in previous work [4] and extend it with allowing for a state dependent soft blocking and capturing the fact that sessions are sometimes dropped. The main assumption that we make is that the session drop probability is connected to the load of the system. When the load is high, the interference from neighbor cells leads to outages with a higher probability than when it is low. For elastic sessions, fast rate and power control attempts to reduce the transmission rates and the required received power at the base station, as long as the transmission rates stay above the session specific so called *guaranteed bit rate* (GBR). Therefore, it seems intuitively clear that there is a trade-off between how conservative the admission control algorithm is (on the one hand) and what is the average bit rate of elastic sessions and what session drop probabilities users experience (on the other hand). The contribution of the paper is to propose a model that can be used for the analysis of this trade-off.

2 Revisiting CDMA Uplink Equations and State Space Structure

The basic CDMA uplink equations that serve as a starting point for this paper are described in details in [3] and [4]. In this section we summarize these results and refer to these references for the derivation of them.

2.1 Revisiting the Basic CDMA Equations

We consider a single CDMA cell at which sessions belonging to one of I service classes arrive according to a Poisson arrival process of intensity λ_i $(i = 1, \ldots, I)$. Each class is characterized by a peak bit-rate requirement $\hat{R}_i$ and an exponentially distributed nominal holding time with parameter μ_i. When sending with the peak rate for a session, the required target ratio of the received power from the mobile terminal to the total interference energy at the base station is given by $\tilde{\Delta}_i = \frac{\hat{R}_i E_i}{W N_0}$. Here E_i/N_0 is the class-wise signal energy per bit divided by the noise spectral density that is required to meet a predefined QoS (e.g. bit error rate, BER) and $W/\hat{R}_i$ is the CDMA *processing gain*.

Let n_i be the number of ongoing sessions of class i. We will refer to vector $\underline{n} = \{n_i\}$ as the *state* of the system. We now assume that arriving sessions are blocked by a suitable admission control algorithm that prevents the system from reaching the state in which the power that should be received at the base station would go to infinity. In other words, a suitable admission control algorithm must prevent the system to reach its *pole capacity* (as defined by Equation (8.10) of [8] and (5) of [3]).

The power P_i that is received at the base station from the mobile terminal for session i must fulfill (see [4]):

$$P_i = \left(P_N + \frac{P_N \cdot \Psi}{1 - \Psi}\right) \cdot \Delta_i = \frac{P_N \cdot \Delta_i}{1 - \Psi}; \Psi \triangleq \Psi(\underline{n}) = \sum_{\ell=1}^{I} n_\ell \cdot \Delta_\ell; \Delta_i \triangleq \frac{\tilde{\Delta}_i}{1 + \tilde{\Delta}_i} \quad (1)$$

Table 1. Model (Input) Parameters

I	Number of service classes
$\hat{R}_i$	Peak bit rate associated with class-i sessions
λ_i	Arrival intensity of sessions belonging to class-i
$1/\mu_i$	Mean (nominal) holding time of sessions belonging to class-i
$\hat{a}_i$	Maximum slow down (using the terminology of [3]) of $\hat{R}_i$
φ	Parameter of the other cell (sector) interference (see Equation (4))
E_i/N_0	Normalized signal energy per bit requirement of class-i

In practice, $\hat{\Psi}$ is defined such that the noise rise in the system stays under some predefined threshold, typically less than 7dB. In the single class case it means that the number of admitted sessions must fulfill: $n_1 < \lfloor \hat{\Psi}/\Delta_1 \rfloor$.

2.2 The Impact of Slow Down

Recall that the required target ratio (Δ_i) depends on the required bit-rate. Explicit rate controlled elastic services tolerate a certain slow down of their peak bit-rate ($\hat{R}_i$) as long as the actual instantaneous bit rate remains greater than $\hat{R}_i/\hat{a}_i$. When the bit rate of a class-i session is slowed down to $\hat{R}_i/a_i$, ($0 < a_i \leq \hat{a}_i$) its required Δ_{a_i} value becomes:

$$\Delta_{a_i} = \frac{\tilde{\Delta}_i}{a_i + \tilde{\Delta}_i} = \frac{\Delta_i}{a_i \cdot (1 - \Delta_i) + \Delta_i}, \quad i = 1, \ldots, I, \tag{2}$$

which increases the number of sessions that can be admitted into the system, since now Ψ_a must be kept below $\hat{\Psi}$, where $\Psi_a = \sum_{i=1}^{I} n_i \cdot \Delta_{a_i}$.

We use the notation $\Delta_{min,i} = \Delta_{\hat{a}_i}$ to denote the class-wise minimum target ratios (can be seen as the minimum resource requirement), that is when the session bit-rates of class-i are slowed down to the minimum value (GBR) associated with that class. The smallest of these $\Delta_{min,i}$ values $\Delta = \min_i \Delta_{min,i}$ can be thought of as the finest "granularity" with which the overall CDMA resource is partitioned between competing sessions.

2.3 Determining the System State Space

The maximum number of sessions from each class can is given by $\hat{n}_i = \lfloor (\Delta_{min,i})^{-1} \rfloor$. Then, recall that in each $\underline{n}$ state of the system, the inequality $\sum_i n_i \cdot \Delta_{a_i} < \hat{\Psi}$ must hold. The states that satisfy this inequality are the *feasible states* and constitute the state space of the system (Θ). The feasible states, in which the acceptance of an additional class-i session would result in a state outside of the state space are the class-i *blocking states*. The set of the class-i blocking states is denoted by Θ_i. Due to the "Poisson Arrivals See Time Averages" (PASTA) property, the sum of the class-i blocking state probabilities gives the (overall) class-i blocking probability. In each feasible state, it is the task of the bandwidth sharing policy to determine the $\Delta_{a_i}(\underline{n})$ class-wise target ratios for each class. The $\Delta_{a_i}(\underline{n})$:s reflect the fairness criterion that is implemented in the resource sharing policy mentioned above. From these, the class-wise slow down factors and the instantaneous bit-rates of the individual sessions can be calculated as follows:

$$a_i(\underline{n}) = \frac{\varDelta_i \cdot (1 - \varDelta_{a_i}(\underline{n}))}{\varDelta_{a_i}(\underline{n}) \cdot (1 - \varDelta_i)}; \quad R_{a_i}(\underline{n}) = R_i / a_i(\underline{n}) \tag{3}$$

For ease of presentation, in the rest of the paper we will not indicate the dependence of a_i, $\varDelta_{a_i}$ and R_{a_i} on the system state $\underline{n}$.

3 Modeling Soft Blocking and Session Drop

3.1 Modeling the Interference from Neighbor Cells

The interference contribution from other cells is typically quite high (around 30-40%). This is taken into account as follows. We think of the CDMA system as one that has a maximum of $\hat{n} = \frac{\hat{\varPsi}}{\varDelta}$ number of (virtual) channels. The neighbor cell interference ξ is a random variable of log-normal distribution with the following mean and standard deviation respectively :

$$\alpha = \frac{\varphi}{\varphi + 1} \cdot \hat{n} \quad \text{and} \quad \sigma = \alpha, \tag{4}$$

where φ is factor characterizing the neighbor cell interference and is an input parameter of the model (Table 1).

The mean value of the interference α is equal to the average capacity loss in the cell due to the neighbor cell interference and σ is chosen to be equal to α as proposed by [6] and also adopted by [5]. (When $\varphi = 0$, the neighbor cell interference is ignored in the model.)

Recall that we think of $\varPsi(\underline{n})$ as the used resource in state $\underline{n}$. Then in a given state $\underline{n}$ let $b_{\varPsi}(\underline{n})$ denote the probability that the neighbor cell interference is greater than the available capacity in the current cell that is ($\hat{\varPsi} - \varPsi$):

$$b_{\varPsi}(\underline{n}) = Pr\{\xi > \hat{\varPsi} - \varPsi\} = 1 - Pr\{\xi < \hat{\varPsi} - \varPsi\} = 1 - D(\hat{\varPsi} - \varPsi),$$

where $D(x)$ is the cumulative distribution function of the log-normal distribution:

$$D(x) = \frac{1}{2}\left(1 + \text{erf}\Big(\frac{\ln(x) - N}{S\sqrt{2}}\Big)\right); N = \ln\left(\frac{\alpha^2}{\sqrt{\alpha^2 + \sigma^2}}\right); S^2 = \ln\left(1 + \frac{\sigma^2}{\alpha^2}\right).$$

The impact of state dependent soft blocking resulted, e.g. by the neighbor cell interference, can conveniently be taken into account by modifying the λ_i arrival rates in each state by the (state dependent) so called passage factor: $\sigma_i(\underline{n}) = g_i(1 - b_{\varPsi}(\underline{n})) = g_i(D(\hat{n} - \varPsi(\underline{n})))$. The passage factor is the probability that a class-i session is not blocked by the admission control algorithm when such a session arrives in system state $\underline{n}$ [5].

3.2 Modeling Session Drop

When the system is in state $\underline{n}$, a class-i session leaves the system with intensity $\gamma_i(\underline{n}) \cdot \frac{\mu_i}{a_i(\underline{n})}$, where $\gamma_i(\underline{n})$ is the state dependent session drop factor. The session drop factor is such that for all i: $\gamma_i(\underline{n})\,|_{n_i=0} = 1;$ and $\gamma_i(\underline{n})\,|_{n_i \neq 0} \geq 1$. Furthermore, we can assume that the drop probability for a given session does not depend on the instantaneous slow down of that session. This is because whether a session gets out of coverage or whether it gets dropped by the radio network does not depend on the slow down. The session drop probabilities, however, depend on the actual level of the noise rise, because higher noise rise level at the base station makes decoding of signals more difficult. We will thus assume that the session drop factor is a function of the macro state only and is the same for all classes : $\gamma_i(x) = f(x) = f(\Psi) \quad \forall i \in I$. That is, we assume that the session drop probability is determined by the load in the system and is equal for all service classes.

4 System Behavior

4.1 The Markovian Property and Determining the Generator Matrix

We now make use of the assumptions that the arrival processes are Poisson and the nominal holding times are exponentially distributed (see Subsection 2.1). The transitions between states are due to an arrival or a departure of a session of class-i. The arrival rates are given by the intensity of the Poisson arrival processes. Due to the memoryless property of the exponential distribution, the departure rates from each state depend on the nominal holding time of the in-progress sessions and on the slow down factor in that state. Specifically, when the slow down factor of a session of class-i is $a_i(\underline{n})$, its departure rate is $\gamma_i(\underline{n})\mu_i/a_i(\underline{n})$. Thus, the system under these assumptions is a continuous time Markov chain (CTMC) whose state is uniquely characterized by the state vector $\underline{n}$.

4.2 Determining the Generator Matrix

For ease of presentation, but without losing generality, we use an example to illustrate the structure of the generator matrix. Assume that $\hat{a}_1 = 1$, $\hat{a}_2 > 1$ and $\hat{a}_3 > 1$. In this case, the task of the bandwidth sharing policy simplifies to determining $\Delta_{a,2}$ and $\Delta_{a,3}$ for each state, from which $\hat{a}_2$ and $\hat{a}_3$ follows.

Based on the considerations of the preceding subsections, we see that the generator matrix $\mathbf{Q}$ possesses a nice structure, because only transitions between ”neighboring states” are allowed in the following sense. Let $q(n_1, n_2, n_3 \rightarrow n'_1, n'_2, n'_3)$ denote the transition rate from state (n_1, n_2, n_3) to state (n'_1, n'_2, n'_3). Then the non-zero transition rates between the feasible states are (taking into account the impact of the passage factors and session drop factors):

$$\begin{aligned}
q(n_1, n_2, n_3 \rightarrow n_1 + 1, n_2, n_3) &= \lambda_1\sigma_1(n_1, n_2, n_3)\\
q(n_1, n_2, n_3 \rightarrow n_1, n_2 + 1, n_3) &= \lambda_2\sigma_2(n_1, n_2, n_3)\\
q(n_1, n_2, n_3 \rightarrow n_1, n_2, n_3 + 1) &= \lambda_3\sigma_3(n_1, n_2, n_3)\\
q(n_1, n_2, n_3 \rightarrow n_1 - 1, n_2, n_3) &= n_1\gamma_1(n_1, n_2, n_3)\mu_1\\
q(n_1, n_2, n_3 \rightarrow n_1, n_2 - 1, n_3) &= n_2\gamma_2(n_1, n_2, n_3)\mu_2/a_2(n_1, n_2, n_3)\\
q(n_1, n_2, n_3 \rightarrow n_1, n_2, n_3 - 1) &= n_3\gamma_3(n_1, n_2, n_3)\mu_3/a_3(n_1, n_2, n_3)
\end{aligned}$$

The first three equations represent the state transitions due to session arrivals, while the second three equations represent the transitions due to session departures. Here we utilized the fact that Class-1 sessions cannot be slowed down, while Class-2 and Class-3 sessions can be slowed $a_2 : 1 \leq a_2 \leq \hat{a}_2$, and $a_3 : 1 \leq a_3 \leq \hat{a}_3$ respectively.

4.3 Determining the Blocking Probabilities and Session Drop Probabilities

From the steady state analysis, the blocking and dropping probabilities directly follow. The hard blocking probabilities can be easily calculated, because we assume that the sessions from each class arrive according to a Poisson process: $P_{hard,i} = \sum_{\underline{n} \in \Theta_i} \pi(\underline{n})$. The total blocking probabilities include the soft blocking probabilities in each state and the hard blocking probabilities: $P_{total,i} = 1 - \sum_{\underline{n} \in \Theta} \pi(\underline{n}) \sigma_i(\underline{n})$. Finally, the class-wise dropping probabilities can be calculated using the following observation. Since the dropping related departure rate from state $\underline{n}$ is $(\gamma_i(\underline{n}) - 1) \cdot \frac{n_i \mu_i}{a_i(\underline{n})}$, the long-term fraction of the dropped sessions must be proportional to $\frac{\gamma_i(\underline{n})-1}{\gamma_i(\underline{n})} \cdot \frac{n_i \mu_i}{a_i(\underline{n})}$. Weighing this quantity with the stationary probability distribution of the system and normalizing yields:

$$P_{drop,i} = \frac{\sum_{\underline{n} \in \Theta} \pi(\underline{n}) \cdot \frac{\gamma_i(\underline{n}) - 1}{\gamma_i(\underline{n})} \cdot \frac{n_i \mu_i}{a_i(\underline{n})}}{\sum_{\underline{n} \in \Theta} \pi(\underline{n}) \cdot \frac{n_i \mu_i}{a_i(\underline{n})}}. \tag{5}$$

In the next section we will show how this intuitively clear formula can be verified by defining a trapping state in this system.

5 Solution Based on the Tagged Customer Approach

The calculation of the (mean and the distribution of the) time to completion of successful sessions requires some additional effort. As we shall see, the method we follow here can also be used to verify the dropping probability calculations as suggested by Equation (5).

5.1 Session Tagging and Modifying the State Space

In order to calculate the moments and the distribution of the holding time of successful (not dropped) sessions we modify the state space by introducing a trapping (absorbing) state and make the following considerations.

We will continue to think of an elastic session as one that brings with itself an exponentially distributed amount of work and, if admitted into the system, stays in the system until this amount of work is completed or the session gets dropped. The method we follow here is based on (1) *tagging* an elastic session arriving to the system, which, at the time of arrival is in one of the feasible states; and (2) carefully examining the possible transitions from the moment this tagged call enters the system until it acquires the required service or gets dropped and therefore leaves the system. Finally, un-conditioning on all possible entrance state probabilities, the distribution of the best effort service time can be determined.

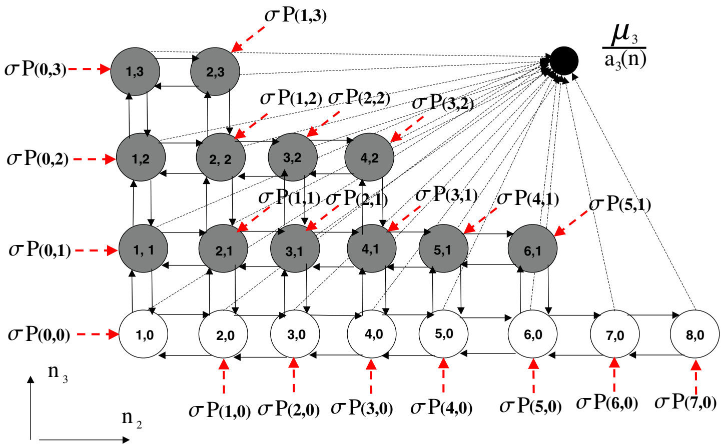


Fig. 1. Modified state space with a trapping state that represents successful session termination. The transition rates to this trapping state correspond to the transition rates with which the tagged session enters the trapping state. The initial probability vector can be determined from the steady state by normalization and taking into account the 'thinning' affect of the passage factors.

For the purpose of illustration, we again concentrate on the part of the state space in which $n_1 = 8$ and tag a class-3 session. Figure 1 shows the state transition diagram from this tagged session's point of view an infinitesimal amount of time after this tagged session entered the system. Since we assume that at least the tagged session is now in the system, we exclude states where $n_3 = 0$. Figure 1 also shows the entrance probabilities for each state, with which the tagged session finds the system in *that* state. Thus, in Figure 1, the tagged arriving session will find the system in state (n_2, n_3) with probability $P(n_2, n_3)$, and will bring the system into state $(n_2, n_3 + 1)$ unless (n_2, n_3) is a Class-3 hard blocking state. For non hard blocking states the entrance probabilities have to be "thinned" with the passage factor (i.e. $\gamma(n_1, n_2, n_3)$). In order for the entrance probabilities to sum up to 1, they need to be re-normalized since we have excluded entrances in the hard blocking states.

In this modified state space, we also define a $trapping\ (absorbing)\ state$. Depending on how this trapping state is interpreted and how the transition rates into that state is defined, we can calculate the moments and the distribution of the holding time of successful sessions and the time until dropping of dropped sessions as well.

We first discuss the case of successful sessions. In this case, the trapping state corresponds to the state which the tagged session enters when the workload is completed ("the file has been transferred successfully"). The transition rates from each state are given by $\mu_3/a(\underline{n})$. The time until absorption corresponds to the time the tagged session spends in the system provided that it is not dropped. Indexing the modified state space in a similar manner as the original state space, the new generator matrix $\tilde{\mathbf{Q}}_\mathbf{S}$ will have the following structure:

$$\tilde{\mathbf{Q}}_\mathbf{S} = \begin{bmatrix} B_S & b_S \\ 0 & 0 \end{bmatrix} \qquad (6)$$

where the B_S matrix represents the transitions between the non-trapping states, the b_S vector contains the transitions to the trapping state, the 0 vector indicates that no transitions are allowed $from$ the trapping state. When the trapping state represents the state that the tagged session enters when it is dropped, the transition rates to the trapping state are given by $\frac{\gamma_3(\underline{n})-1}{a_3(\underline{n})}\mu_3$ and the generator matrix takes the following form:

$$\tilde{\mathbf{Q}}_{\mathbf{D}} = \begin{bmatrix} B_D & b_D \\ 0 & 0 \end{bmatrix} \tag{7}$$

where the B_D matrix represents the transitions between the non-trapping states, and the b_D vector contains the transitions to the trapping state. Once the structure of the expanded state space and the associated transition rates together with the (thinned) initial probability vector, $P_R(0)$, are determined, we can determine the r^{th} moment of T_S:

$$E[T_S^r] = r! \cdot P_R^t(0) \cdot (-B_S)^{-r} \cdot e \tag{8}$$

We note that the procedure to calculate the moments of T_D is the same as that for T_S, except that we now have to make use of the B_D matrix instead of B_S. The distributions of T_S and T_D are given by:

$$Pr\{T_S < x\} = 1 - P_R^t(0) \cdot e^{xB_S} \cdot e; \quad Pr\{T_D < x\} = 1 - P_R^t(0) \cdot e^{xB_D} \cdot e.$$

5.2 Verifying Equation (5): An Alternative Way to Calculate the Dropping Probabilities

The trapping state approach can also be used to determine the dropping probabilities, which can be used to verify results obtained from Equation (5). In order to do this, we consider the modified state space with two trapping states illustrated in Figure 2. From each state, the tagged class-i session can enter any of the two trapping states corresponding to the case when the tagged session successfully terminates or gets dropped. The generator matrix of this state space is given by:

$$\tilde{\mathbf{Q}}_{\mathbf{i}} = \begin{bmatrix} B_i & b_{S,i} & b_{D,i} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9}$$

where $\underline{b}_{drop,i}$ is the column vector containing the transition rates to the trapping state representing the session drops. The B_i matrix has to be determined considering the total transition rates to the two trapping states.

The class-wise dropping probabilities can be calculated using Equation (10):

$$P_{drop,i} = P_R^t(0) \cdot (-B_i)^{-1} \cdot \underline{b}_{D,i}, \tag{10}$$

6 Numerical Results

6.1 Input Parameters

The input parameters for the two cases that we study are summarized by Table 2. In Case I, Class-1 is a rigid class, whereas in Case II, Class-1 is elastic with a maximum slow

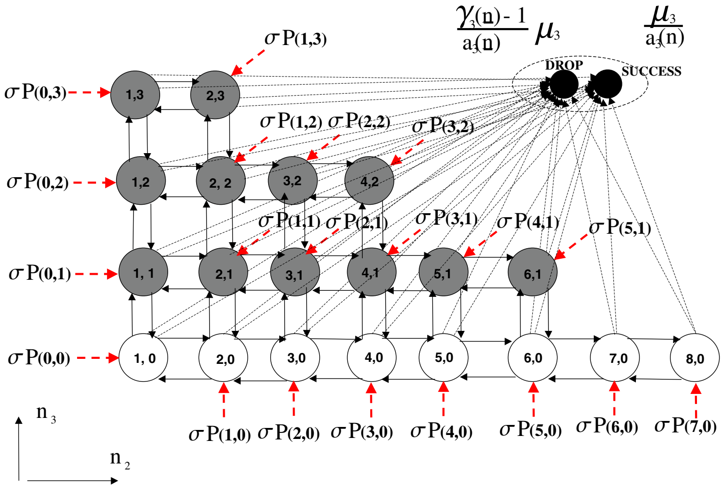


Fig. 2. Modified state space with two trapping states representing successfully terminated and dropped sessions respectively. Seen from the transient states, the total transition rates with which the tagged session enters either of these states is the sum of the two transition rates. This modified state space can be used to determine the probabilities of success and drop.

down factor $\hat{a}_1 = 3$. In both cases we change the maximum slow down factor of Class-2 $\hat{a}_2 = 1 \ldots 4$. ($\hat{a}_2$ is changed along the x axis in each Figure.) The offered traffic is set to 2.72 Erlang per each class and the required Δ_i value for sessions of each class is ≈ 0.15. The function $\gamma_i(\underline{n}) = f(\underline{n})$ is set such that it does not depend on the slow down factors, according to the discussion at the end of Section 3.2. Specifically, in this paper we choose the following dropping factor: $f(\underline{n}) = 1 + \nu \ln(1 + n_1 \cdot \Delta_1 + n_2 \cdot \Delta_2)$, expressing that the dropping factor is a function of the total load in the system (see also Table 2).

6.2 Numerical Results

Blocking Probabilities. Figures 3-4 and Figures 5-6 show the impact of state dependent blocking on the total blocking probabilities. State dependent blocking implies that the admission control takes into account the instantaneous value of the noise rise at the base station rather than just the state of the own cell. This increases the class-wise total blocking probabilities from around 7% and 2% to 10% and 6% in Case I when $\hat{a}_2 = 4$. We also note that when both classes are rigid (Case I, $\hat{a}_2 = 1$), the total blocking values are high, but these high values are brought down to reasonably low blocking probability values when either one and especially when both classes tolerate slowing down of the instantaneous transmission rates (Case II, $\hat{a}_2 = 4$).

Dropping Probabilities. Figures 7-8 and Figures 9-10 show the impact of soft blocking on the session drop probabilities. First, we note that the session drop probabilities slightly (less than 2%) increase as traffic becomes more elastic. The reason is that the system utilization increases when traffic is elastic and the system operates in "higher states" with a higher probability than when traffic is rigid.

Table 2. Model (Input) Parameters

I	2
$\hat{R}_i$	128 [kbps]
λ_i	87.2613 [1/s]
μ_i	32.03 [1/s]
$\hat{a}_1$	1 (Case I); 3 (Case II)
$\hat{a}_2$	1 . . . 4 (along the x axis)
φ	0.25
E_i/N_0	7 [dB]
Dropping factor	$f(\underline{n}) = 1 + \nu\ln(1 + n_1 \cdot \Delta_1 + n_2 \cdot \Delta_2)$,$\nu = 1$; [9]

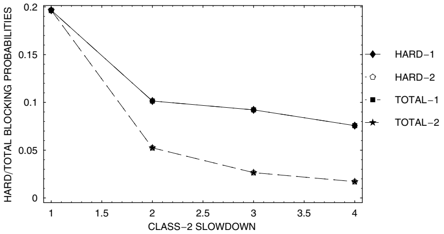


Fig. 3. Case I, no soft blocking, blocking probabilities (total and hard blocking probabilities being equal)

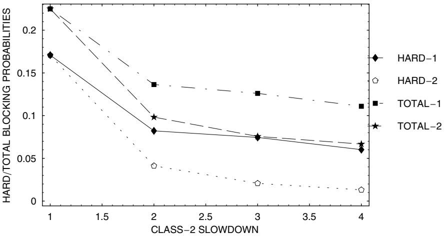


Fig. 4. Case I, soft blocking, blocking probabilities

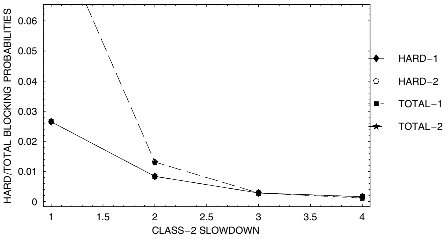


Fig. 5. Case II, no soft blocking, blocking probabilities (total and hard blocking probabilities being equal)

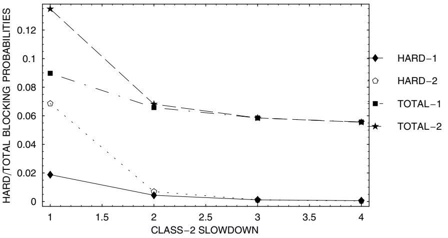


Fig. 6. Case II, soft blocking, blocking probabilities

We also see that state dependent blocking decreases the session drop probabilities in both cases (for example from around 7% to 5% in Case I when $\hat{a}_2 = 4$). This is because soft blocking entails that in average there are fewer sessions in the system that decreases session drops.

Mean Holding Time of the Successful (Not Dropped) Sessions. Figures 11-12 show the mean holding times of successful sessions (normalized to the nominal expected holding time, that is when the slow down factors are 1). In Case I, Class-1 sessions are

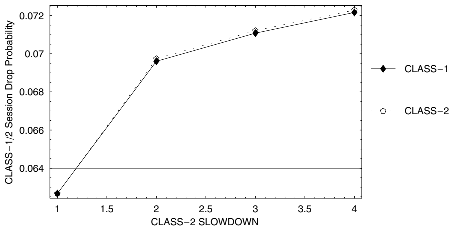


Fig. 7. Case I, no soft blocking, session drop probability

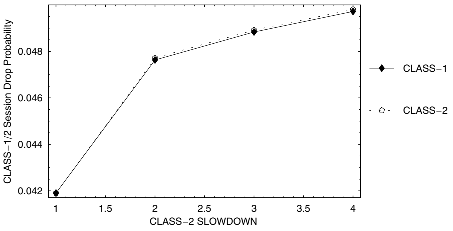


Fig. 8. Case I, soft blocking, session drop probability

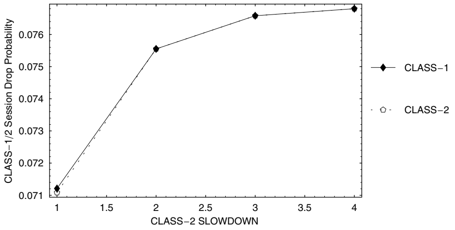


Fig. 9. Case II, no soft blocking, session drop probabilities

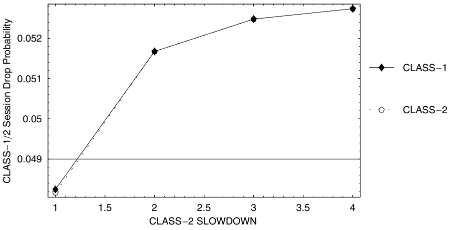


Fig. 10. Case II, soft blocking, session drop probabilities

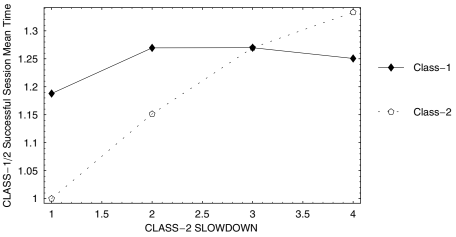


Fig. 11. Case II, no soft blocking, successful sessions' mean holding time

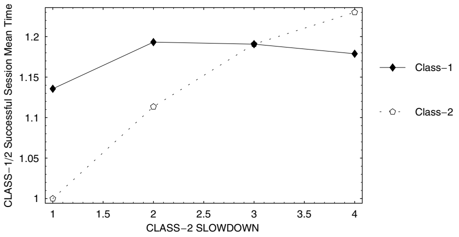


Fig. 12. Case II, soft blocking, successful sessions' mean holding time

rigid and there is no increase in their mean holding times. In this case, Class-2 sessions benefit from soft blocking (keeping in mind that we are now only taking into account the sessions that are successful). Their holding time is somewhat lower in the case of soft blocking.

7 Conclusions

In this paper we have proposed a model to study and analyze the trade-off between the blocking and dropping probabilities in CDMA systems that support elastic services. The model of this present paper captures the impact of state dependent blocking, which is a consequence of the CDMA admission control procedure that takes into account the actual noise rise value at the base station (including the interference coming from surrounding cells) rather than just the state of the serving cell. Session drops happen with a probability that increases with the overall system load.

As traffic becomes more elastic, the session drop probability increases, but this increase can be compensated for by a suitable admission control algorithm. Such state dependent admission control algorithms increase the blocking probabilities somewhat, but this increase can be mitigated if sessions tolerate some slow down of their sending rates. Thus, the design of the CDMA admission control algorithm should take into account the actual traffic mix in the system and the per-class blocking and session drop probability targets.

An important consequence of the presence of elastic traffic is that the blocking probabilities decrease as the maximum slow down factors increase. This is a nice practical consequence of one of the key findings in [3], namely that the Erlang capacity increases. Another consequence of elasticity is that the dropping probabilities increase somewhat, but this increase is not significant (the exact value would depend on the model assumptions, for instance the value of ν).

References

1. A. J. Viterbi, ”CDMA - Principles of Spread Spectrum Communication”, *Addison-Wesley*, ISBN 0-201-63374-4, 1995.
2. J. S. Evans and D. Everitt, ”On the Teletraffic Capacity of CDMA Cellular Networks”, *IEEE Trans. Vehicular Techn.*, Vol. 48, pp. 153-165, No. 1, January 1999.
3. E. Altman, ”Capacity of Multi-service Cellular Networks with Transmission-Rate Control: A Queueing Analysis”, *ACM Mobicom ’02*, Atlanta, GA, September 23-28, 2002.
4. G. Fodor and M. Telek, ”Performance Anlysis of the Uplink of a CDMA Cell Supporting Elastic Services”, in the Proc. of *IFIP Networking 2005*, Waterloo, Canada, Springer LNCS 3462, pp. 205-216, 2005.
5. V. B. Iversen, V. Benetis, N. T. Ha and S. Stepanov, ”Evaluation of Multi-service CDMA Networks with Soft Blocking”, *Proc. ITC Specialist Seminar*, pp. 223-227, Antwerp, Belgium, August/September 2004.
6. A. Mäder and D. Staehle, ”An Analytic Approximation of the Uplink Capacity in a UMTS Network with Heterogenous Traffic”, 18^{th} *International Teletraffic Congress (ITC 18)*, Berlin, September 2003.
7. T. Bonald and A. Proutière, ”Conservative Estimates of Blocking and Outage Probabilities in CDMA Networks” *Performance 2005*, Elsevier Science, June 2005.
8. H. Holma and A. Toskala, ”WCDMA for UMTS - Radio Access for Third Generation Mobile Communications”, Wiley, ISBN 0 471 72051 8, First Edition, 2000.
9. W. Ye and A. M. Haimovich, ”Outage Probability of Cellular CDMA Systems with Space Diversity, Rayleigh Fading and Power Control Error”, *IEEE Communications Letters*, Vol. 2, No. 8, pp. 220-222, August 1999.

A Point-to-Point Protocol Improvement to Reduce Data Call Setup Latency in Cdma2000 System

Eun-sook Lee[1], Kyu-seob Cho[1], and Sung Kim[2]

[1] School of Information & Communication Engineering, SungKyunKwan University, #300,
Chunchun-dong, Jangan-gu, Suwon, Republic of Korea
riya213@skku.edu, kscho103@yurim.skku.ac.kr
[2] Network R&D center, SK Telecom Co., #11, Ulgiro1-ga, Joong-gu, Seoul, Republic of Korea
solar1@sktelecom.com

Abstract. The wireless cellular networks have evolved from IS-95A/B to cdma2000 1X, EV-DO, and WCDMA to improve data service quality. A carries also preserve in their efforts to rev up wireless data service. Despite this efforts, call setup latency has occurred on cdma2000 1X and EV-DO systems. It is an impediment to data service activation. In measuring the real delay time during call establishment from MS to BSC, PCF and PDSN, it is found out that point-to-point protocol between MS and PDSN is major source of delay. Therefore, a simplified PPP is proposed, considering the difference in transmission speed of links among nodes. Then the inter-working scenarios between MS and PDSN is presented with simplified PPP and/or legacy PPP, and the proposed scheme is verified with superior performance over legacy PPP, through comparing the number of packets required for data call setup.

Keywords: Simplified PPP, Call setup latency, Cdma2000 system, Wireless packet data service.

1 Introduction

The paradigm of mobile cellular service has shifted from voice to packet data and a wireless carries have researched and developed a radio technology to provide higher speed data rate, handoff to decrease packet loss rate and wireless TCP considering radio characteristics with burst error. They make efforts to improve an infrastructure to meet a service quality criteria and requirements clarified in ITU-Y.154. It is downstream and upstream transmission rate, packet loss rate, end-to-end packet transmission delay and packet loss rate during handoff on mobile cellular network.

The wireless carriers operating cdma2000 system come to realize that the data call setup time in the system is inferior to General Packet Radio Service (GPRS), and is a key factor in increasing data service usage as well as user friendly UI (user Interface), various contents, billing policy to reduce the tariff burden, and strong security. Users using high speed Internet have not experienced any call setup delay. Even for the data service on the cdma2000 network, these users also want similar level services for data throughput, call setup latency and contents, and so on. When these users attempt the

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 966 – 977, 2006.

wireless data service with their mobile phone, they may abandon the service trial if the call setup latency is much longer than the wired Internet.

Through the measurement and analysis of the delay time from Mobile Station (MS) to Base Station (BS), Packet Control Function (PCF) and Packet Data Serving Node (PDSN) on real network, it is concluded that Point-to-Point Protocol (PPP) operations between MS and PDSN were major source of the delay. A PPP is based on peer-to-peer mechanism is to establish a data link between MS and PDSN. Generally PDSN takes more time to activate the PPP than MS. Because MS can start the PPP as soon as a traffic channel on radio assigns but PDSN has to wait to start it until A8 and A10 session are established. Thus PDSN could lose the first PPP packet from MS. Consequently, this peer-to-peer mechanism requires greater than 1.5 times the number of packets than when two end stations are defined as network and terminal side. In order to solve this problem it is necessary to decrease the number of packets considering the difference in transmission speed of links among nodes.

This paper is organized as follows. In Section II, cdma2000 network model is presented. Section III describes PPP operations for connection establishment. In addition, some of PPP's inherent problems related in call setup latency in cdam2000 system are examined. In Section IV, an improved PPP operation and the state transition diagram to solve setup latency, called simplified PPP (S-PPP), are defined. In section V and VI describe inter-working scenarios between MS and PDSN with S-PPP and/or legacy PPP in cdma2000 system and compare the performances.

2 Cdma2000 Network Model

Figure 1 shows the cdma2000 network model for Simple IP and Mobile IP service. Simple IP refers to a service in which an MS is assigned IP address and is provided IP routing service by an access provider network. Mobile IP refers to a service in which the user is able to maintain a persistent IP address even when handing off between RNs (Radio Networks) connected to different PDSNs [1].

The PPP, which is data link protocol, is located on Medium Access Control/ Link Access Control (MAC/LAC) layer in MS and Generic Routing Encapsulation (GRE) in PDSN. The PPP supports a multiplexing of network protocols and link config-luration, error detection and value-added communication features such as compression

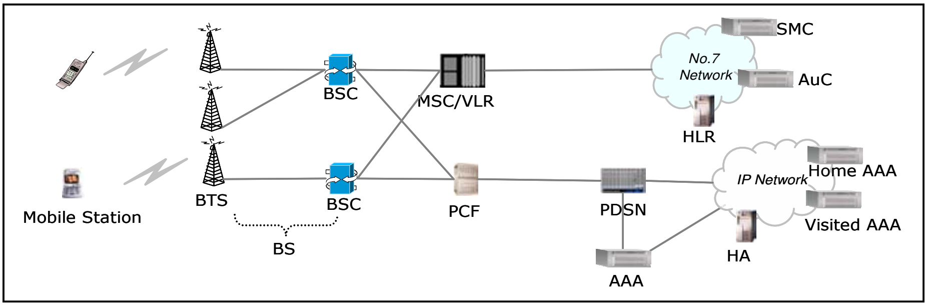


Fig. 1. Cdma2000 Network Reference Model

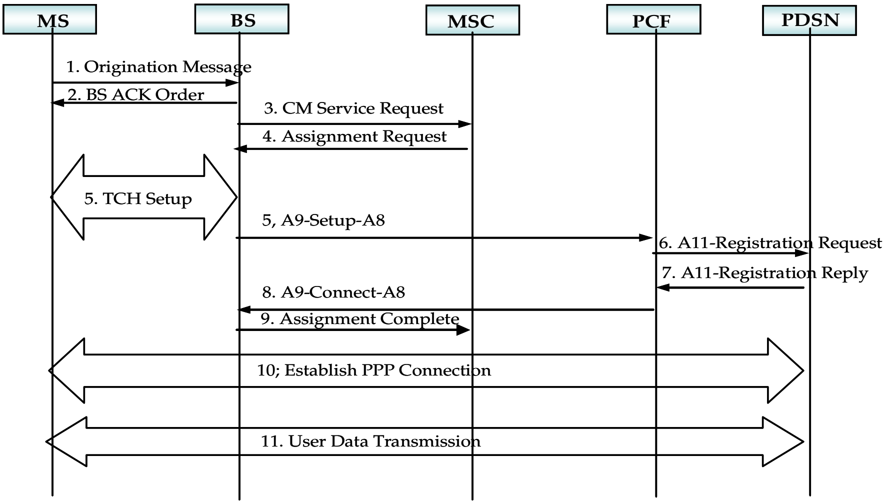


Fig. 2. Cdma2000 Packet Data Service Flows

and encryption, establishing network addresses and authentication [2]. The PDSN assigns IP addresses, transfers it through PPP to MS for Simple IP service and operates as Foreign Agent (FA) for Mobile IP service.

Figure 2 shows call establishment procedure for packet data service initiated by MS in idle and inactive state. The MS sends Origination message to the serving Base Station (BS), which include Service Option, Service instance discriminator and Indicator requesting acknowledgement of layer 2 on an access channel (step 1). The serving BS performs authentication by MSID through the serving MSC interconnec-tion. After positive authentication, the MS assigns radio resource (step 2 to 5). A8 session is established between the serving BS and the PCF using A9 signaling and A10 session between the serving PCF and the PDSN using A11 signaling (step 6 to 9). The procedures of PPP (step 11) are performed in order after completion of A10 session between the MS and the serving PDSN, which are Link Control Protocol (LCP), user authentication and Network Control Protocol (NCP). The MS can send/receive IP packets after PPP establishment (step 12) [3][4].

3 PPP Operations on Cdma2000 Network

The PPP is used widely regardless physical transmission link and transmission rate, and provides compatibility with almost all network technology. Therefore the PPP is utilized on wireless cellular system as link layer protocol between MS and PDSN and is recommended on 3GPP2 ‘wireless IP network’ specification. It would deal with the LCP and IPCP operation included in the PPP in detail and survey the real usage in commercial network [1][2].

3.1 LCP Operations

LCP is to establish, terminate, and maintain data link connections as depicted in Figure 3.

Step 1 & 2. – The serving PDSN sends a Configure-Request packet to the MS to negotiate link configuration after establishment of the A8 and A10 session. The MS also sends a Configure-Request packet to negotiate link configuration after the traffic channel assignment negotiates with BS. The step 2 may be executed prior to step 1. If the serving PDSN receives a Configure-Request before completion of the A10 session, the PDSN discards it.

Step 3 & 4. – The serving PDSN and the MS respond with a Configure-Ack. packet to each peer for reception of a Configure-Request with an acceptable set of configuration options. Upon completion of LCP establishment, the MS and/or the serving PDSN can perform authentication or NCP procedure.

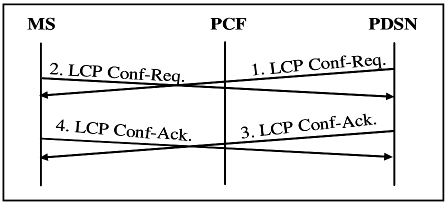


Fig. 3. LCP procedure

3.2 IPCP Operations

The IPCP in RFC 1332, which is one of the NCPs, is the most common and negotiates IP addresses and other parameters to configure, enable and disable the IP network protocol. The IPCP uses the same mechanism as the LCP to exchange packets.

Simple IP service for IPv4. Figure 4a is that the MS requests Simple IP service for IPv4 [1][5].

Step 1& 2. - The serving PDSN sends a Configure-Request packet to the MS with IP Address option for itself after completion of the LCP establishment and positive authentication optionally. The MS also sends a Configure-Request packet to the serving PDSN with IP Address of 0.0.0.0, primary Domain Name System (DNS) address, and secondary DNS address regardless of IPCP state on the serving PDSN.

Step 3. – The serving PDSN assigns the MS an IP address for Simple IP service when the IP Address configuration option in Configure-Request received is a zero. The serving PDSN sends Configure-Nak. packet including new IP address of the MS. It is case that the serving PDSN accepts the DNS address from the MS.

Step 4. **–** The MS acknowledges the Configure-Request with acceptable a specific compression protocol and IP Address configuration option.

Step 5. **–** The MS sends a Configure-Request to the serving PDSN after replacing IP Address option with new IP Address received from the PDSN.

Step 6. **–** The PDSN acknowledges the Configure-Request with acceptable set of-configuration options and transits to an 'Opened' state.

Simple IP service for IPv6. Figure 4b is that the MS requests Simple IP service for IPv6 [1][6].

Step 1. –Because the serving PDSNs do not know the operation that MS requests. Thus, the serving PDSN sends an identical Configure-Request packet to initiate Simple IP service for IPv4 to the MS as in the former case.

Step 2. - The MS sends an IPv6CP Configure-Request packet to request Simple IP service for IPv6 to the PDSN with tentative IF-ID (Interface-Identifier) after completion of the LCP and positive authentication regardless of the PDSN state.

Step 3. – The serving PDSN recognizes the service when receives the Configure-Request's configuration options from the MS and responds with an IPv6CP Configure-Nak. packet including IF-ID to be able to construct the link-local IPv6 address and global IPv6 address at the MS.

Step 4. – The MS sends a Protocol-Reject packet to the serving PDSN and discontinues IPv4CP processing, because the MS of IPv6 mode can not analyze the Configure-Request for IPv4 from PDSN (step 1).

Step 5. – The serving PDSN requests IPv6CP establishment by sending an IPv6CP Configure-Request packet with IF-ID and specific compression protocol.

Step 6. – The MS sends the IPv6CP Configure-Request packet to the serving PDSN after replacing tentative IF-ID option with a new one received from the PDSN.

Step 7. – The PDSN acknowledges for the IPv6CP Configure-Request with acceptable set of configuration options and transits to an 'Opened' state.

Step 8. – The MS also responds with an IPv6CP Configure-Ack. packet for IPv6CP Configure-Request with acceptable set of configuration options to the PDSN and transits to an 'Opened' state.

Mobile IP service for IPv4. Fig. 4c is that the MS initiates Mobile IP service for IPv4 [1][5].

Step 1. – The serving PDSN sends a Configure-Request packet to the MS as the former case.

Step 2. – In order to request Mobile IP service, the MS sends a Configure-Request packet to access network without an IP Address option after completion of the LCP and positive authentication regardless of IPCP state on the serving PDSN.

Step 3. – The serving PDSN acknowledges for the Configure-Request packet with acceptable set of configuration options and transits to an 'Opened' state.

Step 4. – The MS also responds with a Configure-Ack. packet as positive response to the serving PDSN and transits to an 'Opened' state.

After step 4, Mobile IP procedures for mobility management are performed. Both sides of a connection initiate the session establishment shown as dual procedures, because PPP is a peer-to-peer protocol. It might make the whole process unnecessarily complicated. In LCP operations, the MS can send a Configure-Request packet as soon as a traffic channel on radio assigns. The LCP packet could be sent to the serving PDSN before the A8 and A10 session establish, thus it could be lost between BS and PDSN. In this case, the restart timer on the MS expires and then the MS shall retransmit the Configure-Request packet after 3 seconds. Because the default restart timer is 3 seconds. Whenever the timer expires on the MS, the call setup latency becomes longer as much as the timer. The case can be seen several times on the traces of the call setup procedures in the real network. This is because the MS takes a shorter time than PDSN to reach the LCP ready state because of difference in transmission speed of links among nodes.

The purpose of the LCP operation is that the MS wants to set up a bidirectional data link layer session between it and PDSN for packet data services. Considering the roles of the MS and PDSN for its requirements and cddma2000 system characteristics, a simplified asymmetrical process can be designed for the LCP.

The similar arguments might be applied to the IPCP operation. It is excessively complicated because of the peer-to-peer properties and the various kinds of service that should be served. Unnecessary packets are included. For example, the first Configure-Request packet sent by PDSN is useless for IPv6 operation.

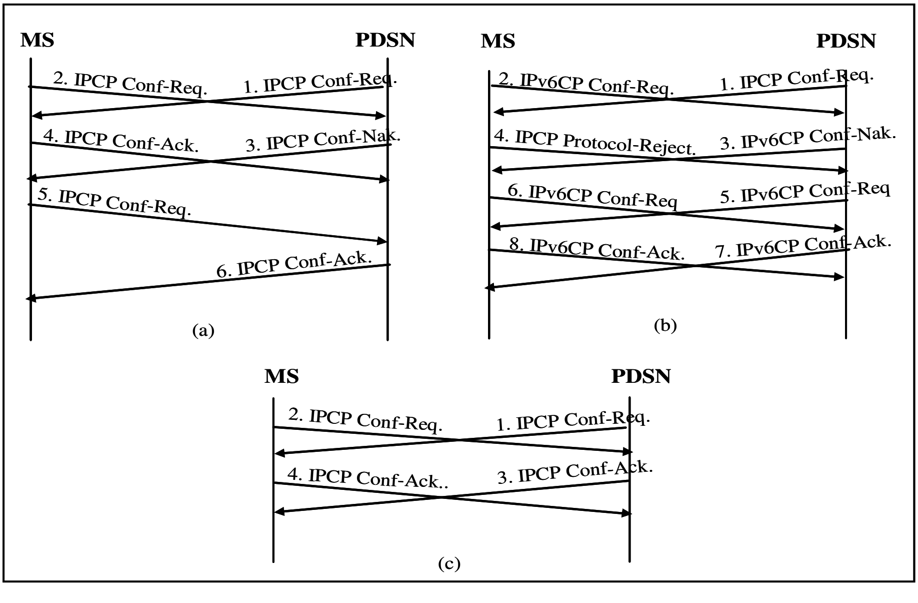


Fig. 4. IPCP procedure for a) Simple IP operation for IPv4; b) Simple IP operation for IPv6; c) Mobile IP operation for IPv4

4 Simplified PPP Proposal

Through the field test measurement and log analysis, simplified PPP (S-PPP) is pro posed for the cdma2000 packet data service. For the packet data service, the MS always initiates the call setup process. PDSN recognizes the call setup request from the MS by receiving A11 signaling. Normally it is PDSN's turn to respond to the request from the MS. If the process is designed based on such request/response call process by network and terminal side definition, not on the peer-to-peer relations, many dual procedures in Figure 3 and 4 can be omitted.

The frame of S-PPP follows High-level Data Link Control (HDLC) format recommended in RFC 1662 and PPP format in RFC 1661[2]. And the operation mechanism for LCP and IPCP state are discriminated network from terminal side to restrict a peer-to-peer characteristic partly like server and client.

4.1 S-PPP Operations

Figure 5 depicts a basic procedure of S-PPP comparing with legacy PPP to establi-sh and terminate a data link between MS and the serving PDSN in normal case.

LCP operations. The step 1 and 2 in Figure 5 conform to LCP operations.

Step 1. – The serving PDSN sends a Configure-Request packet to the MS as previous legacy LCP operation and starts a restart timer. But the MS shall wait until reception of the Configure-Request packet in S-PPP operation. A configuration options in this packet are same with legacy LCP.

Step 2. - The MS responds with a Configure-Ack. packet when it acknowledges the reception of the Configure-Request packet with an acceptable set of configuration options. If the configuration options are not acceptable, the MS selects a configuration options want to negotiate with the PDSN and inserts them in Configure-Reject/Nak. packet and send the packet to the PDSN. If the PDSN can accept the whole configuration options in the Configure-Request packet from the MS, it sends Configure-Request packet with the configuration options and start restart timer. Completion of LCP procedures, authentication is determined during LCP negotiation.

IPCP operations. IPCP operations for Simple IP is from step 4 to 7 and Mobile IP service is step 4 and step 7 in Figure 5.

Step 4. – The MS determines a proper type of packet and a configuration options. In case of Simple IP service for IPv4, the MS sends a Configure-Request packet set the PDSN and MS address in IP Addresses configuration option 0.0.0.0. For Simple IP service for IPv6, the MS sends an IPv6CP Configure-Request packet including a tentative IF-ID. For Mobile IP service for IPv4, the MS sends Configure-Request packet without IP Address or IP Addresses configuration option. Another configuration options are same with legacy IPCP.

Step 5. – The serving PDSN responds with a Configure-Nak. or an IPv6CP Configure-Nak. packet as a negative response, in order to notify the receiving

Configure-Request packet with an unacceptable set of configuration options. Thus the serving PDSN to support Simple IP service includes itself and the MS address in IP Addresses configuration option for IPv4 and an IF-ID configuration option for IPv6.

Step 6. – The MS sends a Configure-Request or an IPv6CP Configure-Request packet again after replacing IP Address or IF-ID with it received from the serving PDSN.

Step 7. – The serving PDSN responds with a Configure-Ack. or IPv6CP Configure-Ack. packet after reception of the Configure-Request or the IPv6CP Configure-Request packet with an acceptable set of configuration options and transits to an 'Opened' state.

Step 8 & 9. – After Completion of successful IPCP negotiation, the PDSN sends an Account-Request packet to Authentication, Authorization and Accounting (AAA) server to notify of a billing start and gets Account-Response from the AAA. And the MS and the PDSN can deliver user data.

Step 10. – When the user wants to stop the service, the MS sends a Terminate-Request packet to the serving PDSN.

Step 11. - The serving PDSN responds with a Terminate-Ack. packet and terminates the S-PPP session. The A10 and A8 session are then release through A11 and A9 signaling.

Step 12 – The serving PDSN sends an Accounting-Request to the AAA server to notify of a billing stop.

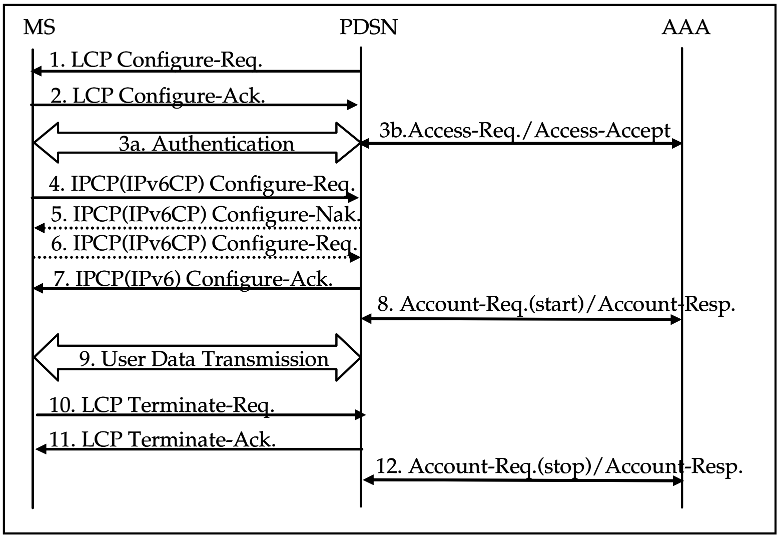

MS
PDSN
AAA
1. LCP Configure-Req.
2. LCP Configure-Ack.
3a. Authentication
3b.Access-Req./Access-Accept
4. IPCP(IPv6CP) Configure-Req.
5. IPCP(IPv6CP) Configure-Nak
6. IPCP(IPv6CP) Configure-Req.
7. IPCP(IPv6) Configure-Ack.
8. Account-Req.(start)/Account-Resp.
9. User Data Transmission
10. LCP Terminate-Req.
11. LCP Terminate-Ack.
12. Account-Req.(stop)/Account-Resp.

Fig. 5. Simplified PPP procedure

4.2 State Transition Diagram of S-PPP

In order to improve the PPP performance, a state transition diagram of S-PPP divides into network and terminal side. Figure 6a depicts the S-PPP state transition diagram on the PDSN for LCP operation and on the MS for IPCP operation. For instance the normal procedure for LCP operation is as follows: The ‘Initial’ state transits to the ‘Req-Sent’ via the ‘Starting’ after an administrative open is initiated and low layer is available and the serving PDSN sends a Configure-Request packet to peer.

After the PDSN receives a Configure-Ack. packet with an acceptable set of configuration options, it transits to an ‘Opened’ state. The ‘Opened’ state means that authentication or IPCP operation can process following the completion of LCP operation and user traffic can be delivered. In IPCP operation, above description is for the MS.

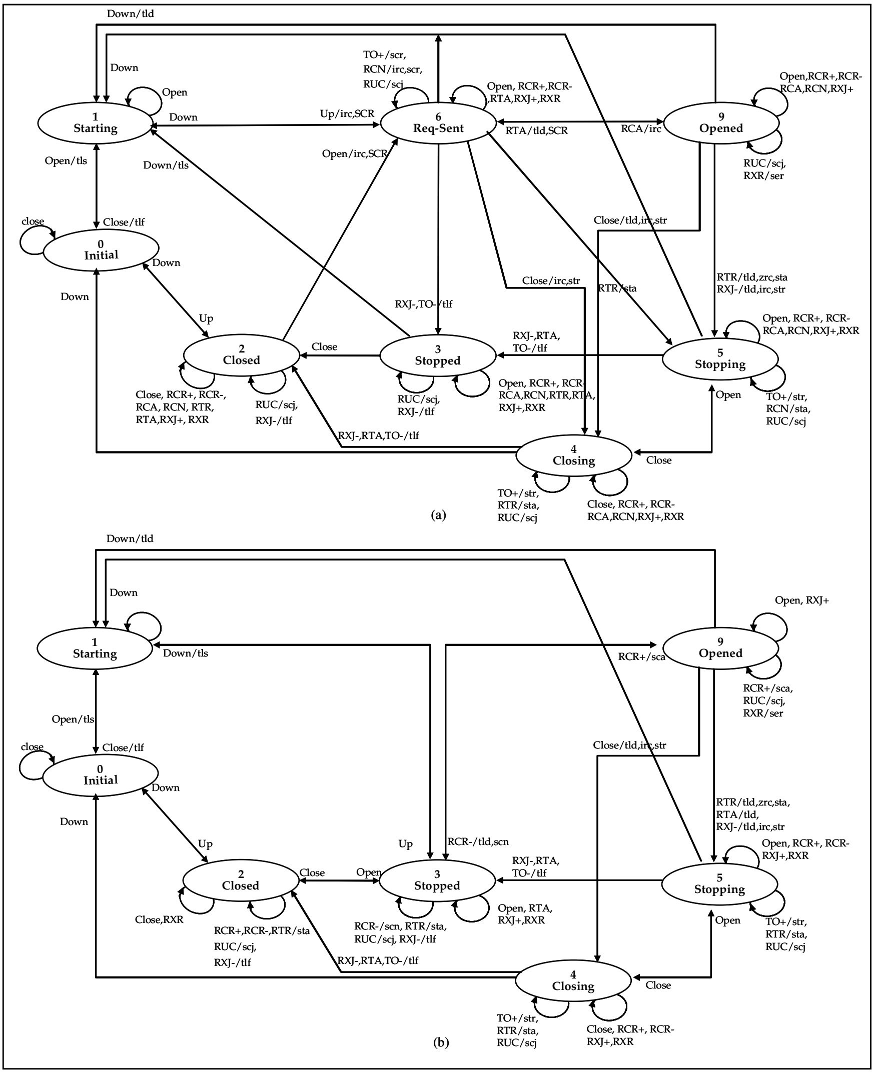


Fig. 6. State transition table for a) LCP operation on PDSN / IPCP operation on MS b) LCP operation on MS / IPCP operation on PDSN

Figure 6b is the S-PPP state transition diagram on the MS for LCP operation and on the PDSN for IPCP operation. For instance the normal procedure for LCP operation is as follows: The 'Initial' state transits to the 'Stopped' state via 'Starting' state after an administrative open is initiated and low layer is available. And the MS waits a Configure-Request packet from peer. After reception of this packet with an acceptable set of configuration options, the MS responds with a Configure-Ack. packet and transits to 'Opened' state. If not, the MS responds with a Configure-Reject or a Configure-Nak. packet and keeps the 'Stopped' state.

5 S-PPP Inter-working Scenarios on Cdma2000 Network

In order to implement the S-PPP on the existing cdma2000 network, the MS and the serving PDSN are required to know how to discriminate a type of PPP, such as legacy PPP and S-PPP, and how to notify a network capability regarding a type of PPP.

It is recommended that the Service Option parameter is able to get through the Origination message as a PPP discriminator. Using the Service Option is a better way to minimize the modification on network nodes and the MS than a new parameter. In the cdma2000 system, the Service Option 33 means packet data service for cdma2000 1X system and 59 for EV-DO. The meaning of Service Option 33 will be expanded for cdma2000 1X packet data service using legacy PPP and 59 for EV-DO using legacy PPP. Thus a new Service Option 80 is defined, which means cdma2000 1X packet data service using S-PPP and the other new Service Option 81 for EV-DO using S-PPP as temporary value. According to the capability of the network and the MS to handle the PPP and/or S-PPP, there are six scenarios to inter-work as follows;

Scenario 1 : The MS and PDSN handle both legacy PPP and S-PPP.
Scenario 2 : The PDSN handles both legacy PPP and S-PPP and the MS handles only legacy PPP.
Scenario 3: The PDSN handles both legacy PPP and S-PPP and the MS handles only S-PPP.
Scenario 4: The PDSN handles only legacy PPP and MS handles both legacy PPP and S-PPP.
Scenario 5: The PDSN and the MS handles only legacy PPP.
Scenario 6: The PDSN handles only legacy PPP and MS handles only S-PPP.

The Scenario 4 and 5 follow the current packet data procedure because the PDSN operates as if it uses legacy PPP. Although the PDSN can inform the MS that it can handle S-PPP in Scenario 2, the MS cannot recognize the situation and it operates as if it was using Service Option 33 or 59, according to network model. In Scenario 6, it is impossible to provide packet data service. Only in the case of Scenario 1 and 3, S-PPP can be activated using Service Option 80 or 81, In order to notify the types of PPP that PDSN can handle, using Over The Air Service Provisioning (OTASP) is proposed [7]. The information carriers want to add to a customer handsets is delivered through SMDPP and OTASP Data Messages in Figure 7. In case of Scenario 1 and 3, the MS gets PDSN capability about the type of PPP through OTASP procedure

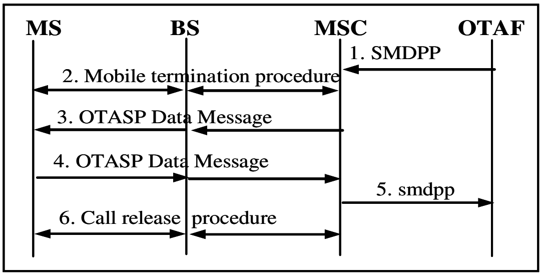


Fig. 7. PPP type notification through OTASP

(Figure 7). When the MS originates a data call on cdma2000, the procedure follows Figure 2 and 5 for S-PPP.

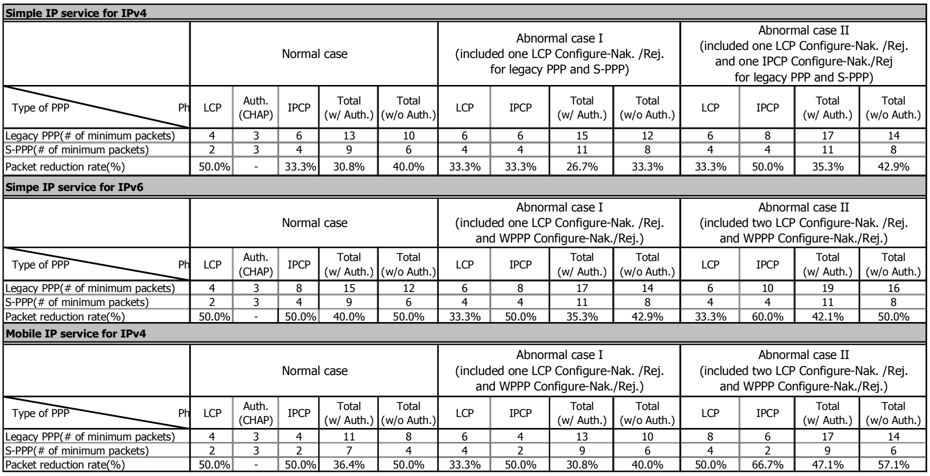
6 Performance Evaluation

The performance is compares by counting the number of packets exchanged between the MS and the PDSN. Table 1 shows the number of packets for a normal case and two abnormal cases in case the authentication is performed and omitted.

First, the normal case is that the legacy PPP operates according to the procedure in Figure 3 and 4 and S-PPP according to the procedure in Figure 5. Second, in abnormal case 1, it is assumed that the MS or the PDSN rejects the Configure-Request for LCP received from peer. Last, abnormal case 2 means that the MS or the PDSN rejects the two Configure-Request packets for LCP and IPCP.

Performing the authentication, 60% to 69.2% of the number of packets for legacy PPP operation are required for the S-PPP operation in the normal case. The packet

Table 1. Comparison of the number of packets between S-PPP and legacy PPP



Simple IP service for IPv4													
	Normal case					Abnormal case I (included one LCP Configure-Nak. /Rej. for legacy PPP and S-PPP)				Abnormal case II (included one LCP Configure-Nak. /Rej. and one IPCP Configure-Nak./Rej for legacy PPP and S-PPP)			
Type of PPP Ph	LCP	Auth. (CHAP)	IPCP	Total (w/ Auth.)	Total (w/o Auth.)	LCP	IPCP	Total (w/ Auth.)	Total (w/o Auth.)	LCP	IPCP	Total (w/ Auth.)	Total (w/o Auth.)
Legacy PPP(# of minimum packets)	4	3	6	13	10	6	6	15	12	6	8	17	14
S-PPP(# of minimum packets)	2	3	4	9	6	4	4	11	8	4	4	11	8
Packet reduction rate(%)	50.0%	-	33.3%	30.8%	40.0%	33.3%	33.3%	26.7%	33.3%	33.3%	50.0%	35.3%	42.9%
Simpe IP service for IPv6													
	Normal case					Abnormal case I (included one LCP Configure-Nak. /Rej. and WPPP Configure-Nak./Rej.)				Abnormal case II (included two LCP Configure-Nak. /Rej. and WPPP Configure-Nak./Rej.)			
Type of PPP Ph	LCP	Auth. (CHAP)	IPCP	Total (w/ Auth.)	Total (w/o Auth.)	LCP	IPCP	Total (w/ Auth.)	Total (w/o Auth.)	LCP	IPCP	Total (w/ Auth.)	Total (w/o Auth.)
Legacy PPP(# of minimum packets)	4	3	8	15	12	6	8	17	14	6	10	19	16
S-PPP(# of minimum packets)	2	3	4	9	6	4	4	11	8	4	4	11	8
Packet reduction rate(%)	50.0%	-	50.0%	40.0%	50.0%	33.3%	50.0%	35.3%	42.9%	33.3%	60.0%	42.1%	50.0%
Mobile IP service for IPv4													
	Normal case					Abnormal case I (included one LCP Configure-Nak. /Rej. and WPPP Configure-Nak./Rej.)				Abnormal case II (included two LCP Configure-Nak. /Rej. and WPPP Configure-Nak./Rej.)			
Type of PPP Ph	LCP	Auth. (CHAP)	IPCP	Total (w/ Auth.)	Total (w/o Auth.)	LCP	IPCP	Total (w/ Auth.)	Total (w/o Auth.)	LCP	IPCP	Total (w/ Auth.)	Total (w/o Auth.)
Legacy PPP(# of minimum packets)	4	3	4	11	8	6	4	13	10	8	6	17	14
S-PPP(# of minimum packets)	2	3	2	7	4	4	2	9	6	4	2	9	6
Packet reduction rate(%)	50.0%	-	50.0%	36.4%	50.0%	33.3%	50.0%	30.8%	40.0%	50.0%	66.7%	47.1%	57.1%

reduction rates are 26.7% to 35.3% in abnormal case 1 and 35.3 % to 47.1% in abnormal case 2.

And omitting the authentication, 50% to 60% of the number of packets for legacy PPP are required for the S-PPP in the normal case. The packet reduction rates are 33.3% to 42.9% in abnormal case 1 and 42.9% to 57.1% in abnormal case 2.

7 Conclusions

The Call setup latency due to PPP becomes an obstacle for packet data service activation on the cdma2000 network. In this paper, an improved PPP is proposed, simplified PPP, as the result of the analysis on the RFC 1661, log file of call setup proce-dure in the real commercial network, and the specification for the cdma2000. In order to simplify the PPP procedure, considering their roles in the network, a simplified asymmetrical process adapts for the LCP and the IPCP operation. The number of packets is also reduced. To demonstrate how the proposed scheme could be applied to the current cdma2000 network, the inter-working scenarios between the MS and PDSN to be able to minimize the system changes are presented. The PDSN notifies the MS of the network capability about the type of PPP using OTASP, and in order to discriminate S-PPP and legacy PPP, a new Service Option value is defined. To evaluate the performance of the S-PPP, the number of packets and between the MS and the PDSN were compared. The S-PPP presents a packet reduction effected greater than 50%.

References

1. 3GPP2, P.S0001-B v2.0. : CDMA2000 Wireless IP Network Standard. Sep., 2004
2. W. Simpson. : The Point-to-Point Protocol (PPP). RFC1661. July 1994.
3. 3GPP2 A.S0016-C v1.0. : Interoperability Specification (IOS) for cdma2000 Access Network Interfaces, Part 6(A8 and A9 Interfaces). Feb., 2005.
4. 3GPP2 A.S0017-C v1.0. : Interoperability Specification (IOS) for cdma2000 Access Network Interfaces, Part 7 (A10 and A11 Interfaces). Feb,. 2005.
5. G. McGrgor. : The PPP Internet Protocol Control Protocol (IPCP). RFC 1332. May 1992
6. D. Haskin and E.Allen. : IP Version 6 over PPP. RFC 2472. Dec., 1998
7. 3GPP2 C.S0016-C. : OTASP of MSs in Spread Spectrum Standards, Release C. Oct., 2004

Performance and Analysis of CDM-FH-OFDMA for Broadband Wireless Systems

Kan Zheng, Lu Han, Jianfeng Wang, and Wenbo Wang

Wireless Signal Processing Lab,
Beijing University of Posts & Telecomms, Beijing, China
zkan@buptnet.edu.cn

Abstract. Frequency-hopping (FH) methods in the Orthogonal frequency division multiplexing access(OFDMA) system,which are to assign user-specific subcarrier to the active users, have been paid much attention to in the broadband wireless communication system. In this paper, we present a novel multiple access scheme, referred to as CDM-FH-OFDMA, which is the extension of FH-OFDMA with code division multiplexing (CDM). This scheme can exploit the frequency diversity gain without the aid of channel coding. And it also can be employed in the multi-cell environment with one frequency reuse factor. Computer simulation demonstrates effectiveness of CDM-FH-OFDMA and the conclusion is followed.

Keywords: OFDM, FH, CDM.

1 Introduction

Orthogonal frequency division multiplexing (OFDM) as a modulation technique is being applied extensively to future wireless broadband systems due to its efficient usage of the available frequency bandwidth and robustness to frequency selective fading environments[1][2]. Meanwhile, code division multiple access (CDMA) has already shown quite a bit of promise in its spectral efficiency through the flexible frequency reuse and multiple access technique for cellular systems [3]. So OFDM combined with code division multiple access (CDMA) has drawn a lot of interests in the research of future mobile communication systems[4][5]. These OFDM-CDMA schemes are derived from the classic DS-CDMA approach, employing mutually orthogonal spreading codes for user separation within one cell, and scrambling codes for distinguishing different cells.

An alternative to using CDMA for multiple-access is to assign user-specific subcarrier to the active users. These schemes are known as OFDMA or frequency-hopping OFDMA[2]. These schemes maintain the user orthogonality also on frequency-selective fading channels, but rely solely on channel coding and interleaving for obtaining the diversity gain. However, it is claimed that these scheme will be more sensitive to inter-cell-interference and therefore not suitable in frequency-reuse-one systems[6].

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 978–989, 2006.

To find out the most appropriate multiple access schemes, it is necessary to investigate the performances of different multiple access schemes not only in the one-cell scenarios but also in the multi-cell scenarios.

In this paper, we propose a novel multiple access scheme, referred to as CDM-FH-OFDMA, which is the extension of FH-OFDMA with code division multiplexing (CDM). Similar to FH-OFDMA, it applies OFDMA for user separation but additionally uses CDM on the data symbols belonging to the same user. The CDM component is introduced in order to achieve additional frequency diversity gain and average the the inter-cell-interference. Like OFDM-CDMA, this CDM-FH-OFDMA exploits the advantages given by the combination of the spread spectrum technique and multi-carrier modulation. Since one user exclusively uses each subset of subcarriers, there is no multiple access interference between different users in the same cell. And the self-interference of one user can be easily decreased by interference cancellation since all superimposed modulated spreading codes of its subcarrier subset are affected by the same channel fading. When considering the cellular system, frequency reuse factor of one can be realized by using different scramble codes in the neighbor cells and inter-cell interference can be avoid by selecting different subcarrier set for the users in the neighboring cells if the system load is not heavy.

This paper is organized as follows. Section 2 gives the brief description of CDM-FH-OFDMA system. The detector structure is described in Section 3. Section 4 describes the simulation configurations, and in Section 5 the simulation results are presented and discussed. Finally, Section 6 gives the conclusion.

Notations: Throughout this paper, matrices and vectors are set in boldface. $()^T$, $()^*$ and $()^+$ denote transpose, conjugate transpose and Moore-Penrose pseudo-inverse, respectively.

2 System Model

Fig.1 shows the block diagram of a CDM-FH-OFDMA system. As the requirement of frequency-hopping method, each frame, which the basic process unit in the system, consists of the N_t OFDM symbols.After the information bits of

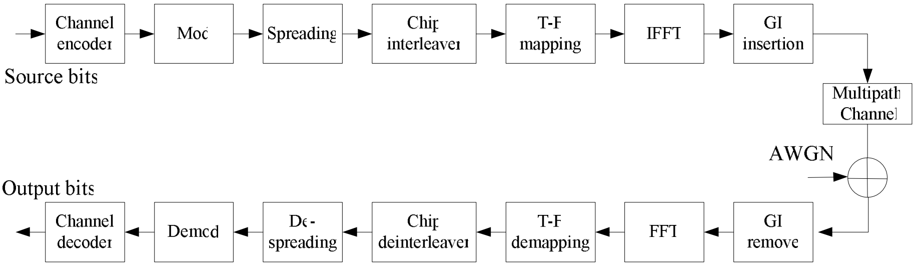


Fig. 1. Block diagram of CDM-FH-OFDMA system

user m are encoded and interleaved, they are modulated to the complex-value symbols with the rate of $1/T_d$. The vector

$$\mathbf{d}_i^{(m)} = [d_{i,0}^{(m)}\ d_{i,1}^{(m)}\ \cdots\ d_{i,Q-1}^{(m)}]^T \tag{1}$$

represents one block of Q parallel modulated data symbols of user m in the ith OFDM symbol.Each data symbol is multiplexed with another orthogonal spreading code of length L. The $L \times Q$ matrix

$$\mathbf{C} = [\mathbf{c}_0\ \mathbf{c}_1 \cdots\ \mathbf{c}_{Q-1}] \tag{2}$$

represents the Q different spreading codes $\mathbf{c}_q = [c_{q,0}\ c_{q,1} \cdots\ c_{q,L-1}]^T \in \mathbb{C}^{L\times 1}, 0 \leq q \leq Q-1$, used by user m, which are the combination of the orthogonal Walsh codes and the base-station-specific scrambling codes. The spreading matrix $\mathbf{C}$ can be same for all the users. The modulated spreading signals are synchronously added, resulting in the transmission vector in the ith OFDM symbol.

$$\mathbf{S}_i^{(m)} = \mathbf{C}\mathbf{d}_i^{(m)} = [S_0^{(m)}\ S_1^{(m)} \cdots\ S_{L-1}^{(m)}]^T \in \mathbb{C}^{L\times 1} \tag{3}$$

To obtain OFDMA scheme, the orthogonal spreading matrix is replaced by the identity matrix.

As shown in Fig.2, in a CDM-FH-OFDMA system, each user may be assigned a specific frequency-hopping sequence that indicates the specific subcarrier subset to use for data transmission in each OFDM block interval,i.e. specific Time-Frequency(T-F) mapping pattern. Multiple data transmissions for multiple users

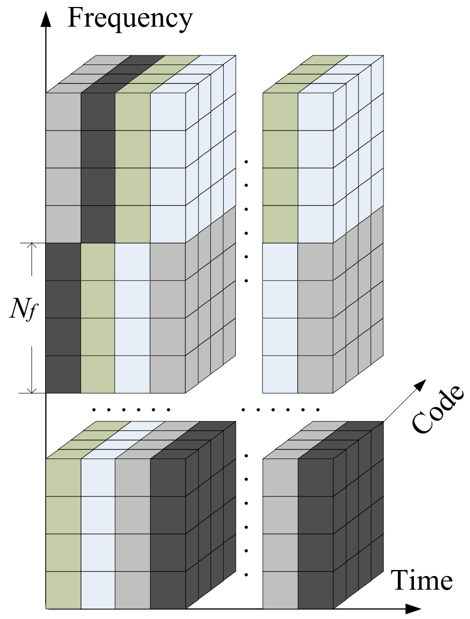


Fig. 2. Multiple access method of CDM-FH-OFDMA system

may be sent simultaneously using different frequency-hopping sequences that are orthogonal to one another, so that only one data transmission uses each subcarrier subset in each OFDM block interval. Using orthogonal frequency-hopping sequences, the multiple data transmissions do not interfere with one another while enjoying the benefits of frequency diversity. For brevity, but without loss of generality, the size of subcarrier subset N_f is assumed to be same as the length of the spreading code,i.e.$N_f = L$. Then, according to the frequency-hopping sequence of user m, the spreading signal of user m is modulated by IDFT onto different subsets of L subcarriers within the N total available subcarriers in one frame ,and the resultant signal in the ith OFDM block interval can be expressed as

$$s_i^{(m)}(n) = \frac{1}{\sqrt{N}} \sum_{l=0}^{L-1} S_l^{(m)} e^{j2\pi(l+P_i^{(m)}L)n/N} \tag{4}$$

where $P_i^{(m)}$ is a hopping index which indicates subcarrier subset indices for the ith OFDM symbol of user m and the T-F mapping pattern for user m is $\mathbf{P}^{(m)} = [P_0^{(m)} \ P_1^{(m)} \cdots \ P_{N_t-1}^{(m)}] \in \mathbb{C}^{1\times N_t}$. Finally the cyclic extension of an OFDM block is added as guard interval before transmission in order to avoid the inter-symbol-interference.

The transmitted signals of different users propagate through independent frequency-selective fading channels. And the channel is modelled as a wide sense stationary, uncorrelated scattering (WSSUS), Rayleigh fading channel with L_t paths and it is assumed that the channel state remains unchanged during at least one OFDM block. Then the channel impulse response during the ith OFDM symbol can be expressed as

$$h^{(m)}(i;n) = \sum_{l=0}^{L_t-1} \alpha_l^{(m)}(i)\delta(n - \tau_l^{(m)}) \tag{5}$$

where the lth tap gain $\alpha_l^{(m)}(i)$ with propagation delay $\tau_l^{(m)}$ of the mth user is independent complex Gaussian random variance with zero mean and variable of $\sigma^2_{m,l,}$.If the cyclic prefix accommodates the channel delay spread between base station and terminals, it is assumed that the narrowband signal that is transmitted through each subcarrier experiences flat Rayleigh fading channel. The flat fading coefficient at the kth subcarrier of the mth user during the ith OFDM block interval can be expressed as

$$H_{i,k}^{(m)} = \sum_{l=0}^{L_t-1} \alpha_l^{(m)} e^{-j2\pi\tau_l^{(m)}k/N}, \tag{6}$$
$$0 \le i \le N_t - 1, 0 \le k \le N - 1$$

After cyclic prefix removal and DFT, the received signals of the desired users can be easily separated according to the hopping pattern and no interference between users exists under the assumption of ideal synchronization since the users are distinguished by an FDMA scheme. The signal vector of user m in the ith OFDM symbol can be written as

$$\mathbf{Y}_i^{(m)} = \mathbf{H}_i^{(m)}\mathbf{S}_i^{(m)} + \mathbf{W}_i = [Y_{i,P_i^{(m)}L}^{(m)} \; Y_{i,P_i^{(m)}L+1}^{(m)} \cdots Y_{i,(P_i^{(m)}+1)L-1}^{(m)}]^T \in \mathbb{C}^{L\times 1} \quad (7)$$

where the $L \times L$ channel fading diagonal matrix for the desired user, AWGN term with zero mean and variance of σ_n^2 are given respectively by

$$\begin{aligned} \mathbf{H}_i^{(m)} &= diag\{H_{i,P_i^{(m)}L}^{(m)}, H_{i,P_i^{(m)}L+1}^{(m)}, \cdots, H_{i,(P_i^{(m)}+1)L-1}^{(m)}\} \\ \mathbf{W}_i &= [W_{i,0} \; W_{i,1} \cdots W_{i,L-1}]^T \in \mathbb{C}^{L\times 1} \end{aligned} \quad (8)$$

3 Detector Structure

Any of the single-user or multi-user detection techniques presented for MC-CDMA systems [4] can be applied for the detection of the data of a single user in CDM-FH-OFDMA systems.

Firstly the received signal vector is equalized by employing a bank of adaptive one-tap equalizers to combat the phase and amplitude distortions caused by the mobile fading channel on the subcarriers. The one-tap equalizer is simply realized by one complex-valued multiplication per subcarrier. The received sequence at the output of the equalizer in the ith OFDM block interval can be expressed as

$$\mathbf{Z}_i^{(m)} = \mathbf{G}_i^{(m)}\mathbf{Y}_i^{(m)} \quad (9)$$

The diagonal equalizer matrix

$$\mathbf{G}_i^{(m)} = diag\{G_{i,0}^{(m)}, G_{i,1}^{(m)}, \cdots, G_{i,L-1}^{(m)}\} \in \mathbb{C}^{L\times L} \quad (10)$$

represents the L complex-valued equalizer coefficients of the subcarriers assigned to $S_i^{(m)}$. The complex-valued output $\mathbf{Z}_i^{(m)}$ is despread by correlating it with the spreading matrix $\mathbf{C}$. The complex-valued soft-decided values at the output of the despreader is

$$\tilde{\mathbf{d}}_i^{(m)} = \mathbf{C}^T\mathbf{Z}_i^{(m)} = [\tilde{d}_{i,0}^{(m)} \; \tilde{d}_{i,1}^{(m)} \cdots \; \tilde{d}_{i,Q-1}^{(m)}]^T \in \mathbb{C}^{Q\times 1} \quad (11)$$

The data symbol in the hard-decided detected vector $\hat{\mathbf{d}}_i^{(m)} = [\hat{d}_{i,0}^{(m)} \; \hat{d}_{i,1}^{(m)} \cdots \hat{d}_{i,Q-1}^{(m)}]^T \in \mathbb{C}^{Q\times 1}$ is given by

$$\hat{d}_{i,q}^{(m)} = \mathit{\Gamma}\{\tilde{d}_{i,q}^{(m)}\}, 0 \le q \le Q-1 \quad (12)$$

where $\mathit{\Gamma}\{*\}$ is the quantization operation according to the chosen data symbol alphabet.

Several different diversity combining techniques have been proposed in the literature. In this paper the equalizer according to the minimum mean error square(MMSE) is used in the despreading by choosing the equalization coefficients as

$$G_{i,l}^{(m)} = \frac{H_{i,P_i^{(m)}L+l}^{(m)*}}{Q|H_{i,P_i^{(m)}L+l}^{(m)}|^2 + \sigma_n^2}, 0 \leq l \leq L-1 \tag{13}$$

The detector described above is originated from the principle of single-user detection (SD), which only detects the desired signal without taking into account any information about multi-code interference. With the number of spreading vectors (i.e. Q) is increased, the performance of such SD detector will be deteriorated due to more serious multi-code interference. The interference cancellation (IC) can improve the performance of the system with heavy load at the expense of higher receiver complexity. The principle of parallel interference cancellation (PIC) is introduced[7] and can also be applied to CDM-FH-OFDMA systems. The basic idea is to take the estimated data obtained by the initial SD detector and regenerate the transmitted signals to calculate the occurring multi-code interference of each desired data signal caused by all other spreading signals, then to subtract it from the received signals. When only the qth,$0 \leq q \leq Q-1$,transmitted data symbol $d_q^{(m)}$ of user m is desired and others are regarded as interference,the corresponding signal after interference-reduced can be written as

$$\mathbf{Y}_{i,q}^{(m)} = \mathbf{Y}_i^{(m)} - \mathbf{H}_i^{(m)}\mathbf{C}_q^{-}\mathbf{d} \tag{14}$$

where $\mathbf{C}_q^-$ denotes the modified spreading matrix obtained by zeroing the qth column of $\mathbf{C}$.

After this cancellation step, the next data detection including equalization and despreading/combining are applied to the interference-reduced signal $\mathbf{Y}_{i,q}^{(m)}$, leading to more reliable data estimations than the initial SD detection. The equalization coefficients after interference-reduction are changed to

$$G_{i,l}^{(m)} = \frac{H_{i,P_i^{(m)}L+l}^{(m)*}}{|H_{i,P_i^{(m)}L+l}^{(m)}|^2 + \sigma_n^2}, 0 \leq l \leq L-1 \tag{15}$$

In each stage, Q interference cancellation and data estimation will be performed. This iterative procedure can be continued until the performance is satisfied.

4 System Configuration

The key simulation parameters are summarized in Table 1 and are kept in same with [2] in order to be compatible with FH-OFDMA system, which has been widely studied in 3GPP long-time evolution.The selective fading channel models including PB3, VA120 defined by ITU are used in the simulations[9]. Each path of the channel is modelled as a classical Jakes Doppler spectrum. Under the assumption of a quasi-stationary channel, the channel is constant during one OFDM interval. At the receiver, perfect symbol/carrier synchronization and channel state information are assumed to be available.

Table 1. System Parameters

Carrier	2GHz
Frame duration (ms)	2
DFT size	1024
Cyclic Prefix interval (samples/ μs)	64/9.803
Subcarrier separation (kHz)	6.375
OFDM block duration (μs)	166.67
Number of OFDM symbols per frame N_t	12
Number of useful data subcarriers	600
Size of data subcarrier subset	40
Number of spreading codes	8
Channel coding/Decoding	Turbo codec/ MAX_LOG_MAP (iteration=8)
Channel model	PB3, VA120 [9]
Spreading factor (Q)	8
Modulation	QPSK

Table 2. Information bit payload and code block sizes for transport format

Modulation	Code Rate	Information bit payload	24-bit-CRC addition	Code Block Segmentation	R=1/3(K=4) Turbo-coding	Rate matching
QPSK	1/2	480	504	1×504	1524	960
QPSK	2/3	640	664	1×664	2004	960
QPSK	4/5	768	792	1×792	2388	960

4.1 Interleaving

The interleavers have to be applied in the CDM-FH-OFDMA systems in order to better explore the diversity gain inherent in the time-frequency selective fading channel. There are two positions that the interleaving will be performed at the transmitter. One is before modulation module, i.e. bit-interleaving, the other is before the IDFT,i.e. chip-interleaving. Since all the data symbols are processed according to the frame unit in the time domain, the interleaving will be applied within one frame including N_t OFDM symbols.

In the transmitter, the binary information is first coded with CRC attachment and code block segmentation before channel coding. The coded bit is punctured and bit-interleaved according to [8]. Table 2 provides appropriate information bit payload and code block segmentation values for the test cases in the simulations. For the sake of implication,only one transport block size for each user is assumed in this paper. Other sizes may also be evaluated if necessary.

A block interleaver is applied in case of chip-interleaving, which is a matrix with N_{depth} rows and N_{length} columns. The symbols are written into the matrix by rows and read out afterward by columns. The deinterleaver puts the symbols into a matrix with the same size, but the symbols are written by columns and read out by rows. Usually the number of rows N_{depth} is defined as the interleaving depth while the number of the column N_{length} as the interleaving length. In the simulations, the N_{depth} and N_{length} of the block chip-interleaving equal to the size of the subcarrier subset and the number of OFDM symbol per frame,respectively.

4.2 Frequency-Hopping Pattern and Spreading

The different users are distinguished by allocating each user with a separate pattern for the time-frequency(T-F) mapping of OFDM units. All the T-F mapping patterns in a cell should be orthogonal to avoid the cross-interference between different users.Each of them should provide not only a maximized diversity gain within a cell but also a minimized inter-cell-interference between the neighboring cells [2].

Therefore,the set of 15 orthogonal T-F pattern, one for each user, is derived from a single generic Costas sequence of length 15. Since the number of OFDM symbol per frame is 12 (i.e. $N_t = 12$), the right T-F pattern of length N_t is obtained by discarding the last three symbols of the generic Costas sequence. Then, the first T-F mapping pattern is given by

$$\mathbf{P}^{(0)} = [13\ 5\ 3\ 9\ 2\ 14\ 11\ 15\ 4\ 12\ 7\ 10] \tag{16}$$

All the T-F pattern in the set are obtained from the first patten in the set by all the different cyclic shifts in the frequency domain. In that way, it is ensured that the set of pattern is orthogonal.

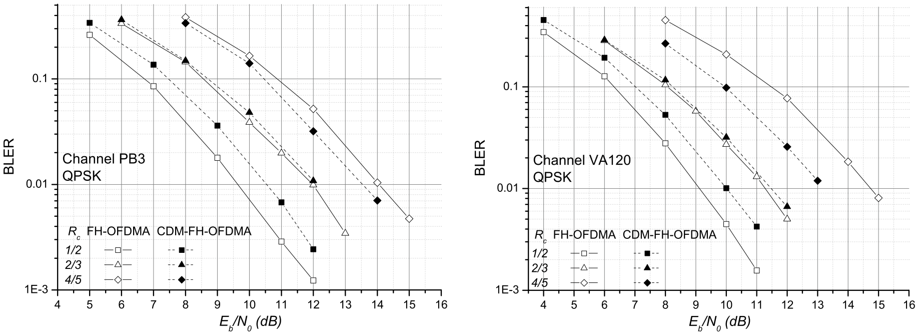
All the time-frequency mapping patterns in a frame are cyclically time-shifted by a cell-specific offset, corresponding to an integer number of OFDM symbols. The time offset is changed for each frame, according to a cell-specific scrambling sequence. In that way, even if the two cells are synchronous in one frame, they will very probably be asynchronous in the next frame, resulting in a minimized cross-interference, as predicted by the correlation properties of time-frequency mapping patterns.

In the simulated CDM-FH-OFDMA system, the data subcarrier subset size N_f is larger than the spreading code length L. So there are N_f/L spreading signal of length L within one data subcarrier subset of size N_f.

5 Performance Evaluation

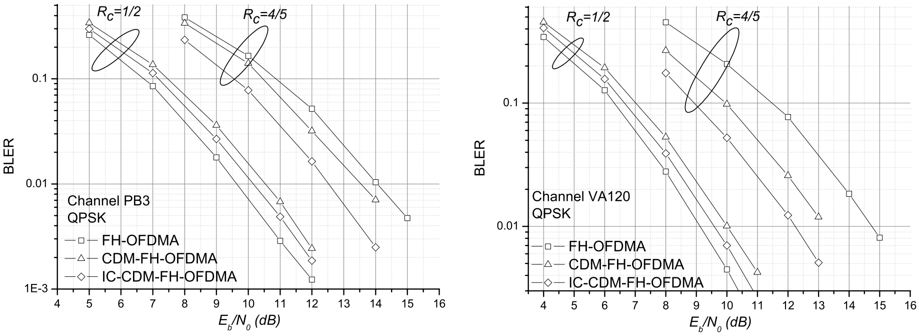
5.1 Single-Cell

In Fig.3(a) and Fig.3(b), the performances of the proposed CDM-FH-OFDMA and FH-OFDMA system applying MMSE principle with the different code rates under PB3 channel or VA120 channel are compared, where QPSK modulation is used. In case of low or medium code rate, the BLER performances of FH-OFDMA are equivalent to or better than these of CDM-FH-OFDMA because it can well exploit the frequency diversity gain through channel coding and



(a) PB3 (b) VA120

Fig. 3. BLER performances in FH-OFDMA or CDM-FH-OFDMA using MMSE detector



(a) PB3 (b) VA120

Fig. 4. BLER performances in FH-OFDMA or CDM-FH-OFDMA using PIC detector

the multi-code interference in CDM-FH-OFDMA deteriorates the BLER performance. However, with the high code rate, the CDM-FH-OFDMA system can make better use of the frequency diversity gain by the operations of spreading and combination than the channel coding. For example, if the target BLER is assumed to be 10^{-2}, the BER gain of CDM-FH-OFDMA compared with FH-OFDMA is about 1 or 2dB under PB3 channel or VA120 channel in case of $R_c = 4/5$.

Fig.4(a) and Fig.4(b) and compare the performances of the proposed CDM-FH-OFDMA and FH-OFDMA system applying MMSE principle or PIC detector with the different code rates under PB3 channel or VA120 channel, where QPSK modulation is used. When the code rate is low(e.g. $R_c = 1/2$), the performance of CDM-FH-OFDMA system are mainly affected by the channel coding/decoding and the interference cancellation won't achieve much performance gain. Then, the BLER performance of CDM-FH-OFDMA system with PIC still keeps little worse than that of FH-OFDMA system. On the other hand, with the higher code rate(e.g.$R_c = 4/5$), the PIC detector will give about 1dB gain if the target BLER is assumed to be 10^{-2} under both channel environments.

5.2 Multi-cell

One of the advantages of CDM-FH-OFDMA system is that the same bandwidth can be reused in each cell, which is often refereed to a full frequency re-use, or frequency re-use factor of 1. The main benefit of such a frequency reuse is mainly ease of deployment, given that no frequency planning is required. However, the CDM-FH-OFDMA system with a frequency reuse of 1 becomes interference-limited,and the interference perceived by the terminal from the different cells might not be perfectly white. First, the interfering signals undergo time dispersion, and hence, do not have a flat spectrum. Furthermore, if the OFDM units are not all being used in the interfering cells, the resulting spectrum from each of these partially-load interfering cells will contain gaps. It is therefore likely that the total interference spectrum observed by the terminal would not be flat, and hence, it might not be accurately modelled using white noise.

The impact of realistic inter-cell interference on performance has therefore been evaluated using the multi-cell simulator with a radio geometry concept, which consider relative inter-cell interference and independent fading for a limited number of chip-exact modelled terminal in a neighboring cell[10]. To become independent from absolute path gains as well as cell layouts, we use the geometry factor which is defined as

$$G = \frac{I_{or}}{I_{oc,b} + I_{oc}} \qquad (17)$$

where I_{or} denotes the received total power originated from the serving, I_{oc} is the received total power from the all the interfering cells and $I_{oc,b}$ is the portion of the received power from those cells modelled as AWGN. Considering the trade-off between the reliability and simulation complexity,the inter-cell-interference is assumed to be generated by a single neighboring cell. It is generated by using the same time-frequency mapping patterns as in the serving cell. Namely, in all cells, the time-frequency mapping patterns in a frame are cyclically time-shifted by a cell-specific offset, corresponding to an integer number of OFDM symbols. The time offset is changed for each frame, according to a cell-specific scrambling sequence.

For the sake of simplification, there is only one user in the serving cell and the inter-cell interference dominates over the noise in this simulation scenario. It is assumed that 100% of the additive interference plus noise is due to the inter-cell-interference and 0% due to the thermal noise, i.e.,

$$\frac{I_{oc}}{I_{oc,b} + I_{oc}} = 1 \qquad (18)$$

The simulation results with the different cell load in the interfering cell are plotted in Fig.5,which gives the performances of FH-OFDMA and CDM-FH-OFDMA with MMSE detection. Less geometry factor, further distance to the

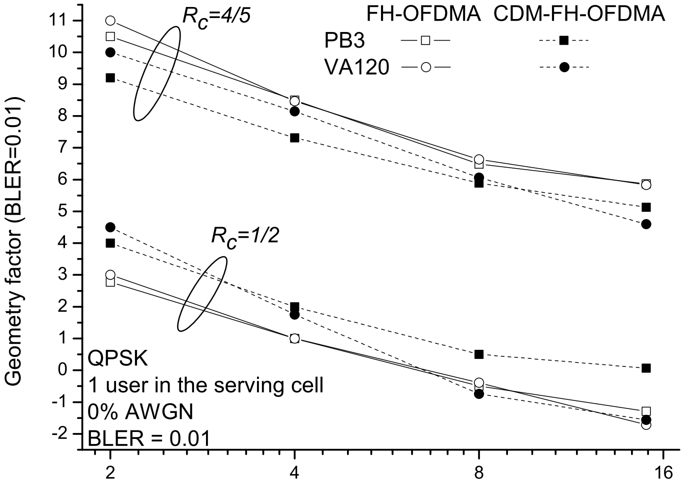


Fig. 5. Required geometry factor with BLER=10^{-2} under different cell load in the interfering cells

base station the user is. When the cell load in the interfering cell increases, the link-performance improves. This perhaps somewhat counterintuitive result is explained as follows.When the users in the interfering cells increases, the collision probability of the subcarriers between the serving user and the interfering users becomes larger. So the interference becomes more like Gaussian according to the center-limited principle. Meanwhile, the signal power of each interfering users decreased with the number of users increases in order to keep the total interference power constant. Therefore, the required geometry factor with BLER=10^{-2} decreased with the number of the users in the interfering cell,which means the serving user can get the same BLER performance in the further position to the centering base station.

In case of low code rate, the performances of FH-OFDMA are little better than those of CDM-FH-OFDMA because the channel coding can exploit the diversity gain well,which is similar to the single-cell case. However, with the code rate increases, the proposed CDM-FH-OFDMA system with the simple MMSE detection outperforms FH-OFDMA system.

6 Conclusion

In this paper,we propose a CDM-FH-OFDMA scheme for the downlink high data rate transmission in the broadband wireless communication system. Simulation results demonstrate that the performances of CDM-FH-OFDMA system are better than those of conventional FH-OFDMA in case of medium or high code rate. Also, the interference cancellation can be applied in CDM-FH-OFDMA systems to further improve the performance.

References

[1] Eklund, C.; Marks, R.B.; Stanwood, K.L.; Wang, S,"IEEE standard 802.16: a technical overview of the WirelessMAN air interface for broadband wireless access ," *IEEE Communications Magazine*,vol.40,no.6,pp.98-107,2002

[2] 3GPP TR 25.892. V2.0.0. (2004-06).Feasibility Study for OFDM for UTRAN enhancement. (Release 6)

[3] H. Holma and A. Toskala, *WCDMA for UMTS*. New York: Wiley, 2000.

[4] S .Hara,R .Prasad, "Overview of multicarrier CDMA," *IEEE Communications Magazine*,vol.35,no.12,pp.126 - 133,Dec. 1997

[5] K. Zheng,G. Zeng,W .Wang,"Performance Analysis for OFDM-CDMA with Joint Frequency-time Spreading," *IEEE Transactions on Broadcasting*, vol.51, no.1, pp.144- 148,March 2005

[6] Maeda, N.; Atarashi, H.; Abeta, S.; Sawahashi, M.;"Throughput comparison between VSF-OFCDM and OFDM considering effect of sectorization in forward link broadband packet wireless access," in *Proc.IEEE Vehicular Technology Conference*, vol.1, pp.47-51,Fall. 2002

[7] M.K.Varanasi, B.Aazhang, "Multistage detection in asynchronous code-division multiple-access communications," *IEEE Trans. Commun.,*vol.38,no.4,pp. 509 - 519, April 1990

[8] 3rd Generation Partnership Project (3GPP), 3G TS 25.211, v3.5.0, *Physical Channels and Mapping of Transport Channels onto Physical Channels (FDD)*, Dec. 2000.

[9] Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000, Rec. ITU-R.M1225.

[10] Weber, R.; Schulist, M.; Schotten, H."WCDMA multi-cell link-level performance," *Proc.IEEE Personal, Indoor and Mobile Radio Communications*, vol.3,pp.1362 - 1366,Sept. 2002

Multi-service Routing: A Routing Proposal for the Next Generation Internet

António Varela[1], Teresa Vazão[2], and Guilherme Arroz[1]

[1] Instituto Superior Técnico, Portugal
antonio.varela@tagus.ist.utl.pt
[2] Inesc-ID,Portugal

Abstract. Quality of Service support plays a major role in the Next Generation Internet. QoS routing protocols must cope with service differentiation to enhance this support. This paper proposes a service aware QoS routing protocol, the Multi-Service routing, which is an extension to traditional intra-domain routing protocols. It proposes a new path selection policy that guides higher priority traffic through the shortest path and diverts lower priority traffic through longer paths when service performance degradation is foreseen. Simulations results shows that the proposed routing performs better than existing QoS routing and link-state protocols.

1 Introduction

Quality of Service (QoS) plays a major role in the Next Generation Internet (NGI), as new services and applications arise based on multimedia traffic with special requirements, demanding new service models and routing approaches [1].

The Internet Engineering Task Force (IETF) attempts to solve Internet's lack of QoS, by defining new services models. The first model proposed - Integrated Service (IntServ)provides strict QoS guarantees, but does not scale well to large networks. The Differentiated Service (DiffServ)model solved this issue and is able to assure QoS to aggregated traffic flows classified into a restricted set of service classes. Multi-Protocol Label Switching (MPLS)solution, which assures QoS support by means of traffic engineering capabilities offered below the network layer. Concerning the QoS all these technologies are expected to coexist on the NGI.Nevertheless, Diffserv will play a central role, as it offers a scalable network layer solution, being then independent of any kind of access technology or higher layer protocols.

To date, the Internet routing focuses on connectivity: routing protocols, such as the Open Shortest Path First (OSPF) or the Routing Information Protocol (RIP), are able to cope with the network impairments, but are unable to fulfill the service requirements imposed by the new kind of applications, being inadequate for the NGI. Traffic between two end points is forwarded through the same path, which is usually the shortest one, disregarding the network conditions and the QoS requirements of the associated flows. Thus, congestion arises in these

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 990–1001, 2006.

paths and service requirements can no longer be met, despite the existence of alternative underutilised paths.

Several QoS aware routing protocols have been proposed to solve these issues [2]. Should data and telecommunication networks converge around the NGI, the QoS routing problems will become very difficult to solve. First of all, this convergence leads to the existence of traffic with diverse QoS constraints in the same network and, according to [3], this may increase routing's complexity, as finding a feasible path with two independent constraints is an NP complete problem. Second, as the network state changes very often it may be difficult to gather up-to-date state information, specially in large scale environments. The use of outdated information by a routing protocol may degrade the network performance. And finally, a network where resources are shared among priority and Best Effort (BE) traffic is difficult to manage. Although performance guarantees can be assured in priority traffic, by means of resource reservation, the throughput of BE traffic will suffer, if the network capacity is under optimised, by wasting paths that may be used at least by BE traffic. Most of the QoS routing proposals are able to deal with the network state' information, but do not cope with service differentiation.

This paper aims at defining a new approach to intra-domain QoS support, where the routing protocol cooperates with DiffServ. The proposal is targeted at IPv6 networks and complaint to MPLS traffic engineering mechanisms, being particularly foreseen to the NGI.

The paper is organized as follows: section 2 presents several approaches for QoS routing; section 3 describes the routing architecture; section 4 contains the simulation results and, finally, section 5 presents the conclusions and future work.

2 QoS Routing in the NGI

NGI QoS routing's support three main tasks: state maintenance, route calculation and path selection. The next sections analyse several possible approaches.

2.1 State Maintenance

State Maintenance is supported by local measurements that are performed at each node to evaluate its own state, regarding a single or multiple performance indicator. It can comprise link occupancy, residual bandwidth, delay or the availability of other resources.

A **Local State** strategy is used whenever each node only uses the information it gathers to compute the routes. Nclakuditi et *al* [4] uses such approach by selecting the path, that will be used to forward a flow, among a set of candidate ones, based on local information. Despites its simplicity, routing decisions are based on an inaccurate view of the network, as remote network conditions are not known.

Should local state information be disseminated through the network, a **Global State** strategy will be used. Although the network state changes very often, routing updates should be bound to reflect the longterm behaviour of the network.

Thus, instead of advertising instantaneous performance indicators, quantified metrics must be used. A simple solution was proposed within the ARPANet scope and consists in calculating the average value of the performance indicator [5]; alternatives are also used based on threshold values and hysteresis mechanisms [6]that reduce routing instability and limits the burden of traffic and processing entailed by the routing protocol.

The complexity of this Global State strategy may be compensated by the most accurate view of the network state that can be achieved when compared to the perspective attained by the Local State strategy. However, in large scale networks a less precise view of the network is accomplished, as longer delays are expected to disseminate and update the routing information. Lack of scalability also arises when the number of metrics to be advertised grows beyond a certain limit. A hybrid strategy based on **State Aggregation** can be used, where nodes are organised hierarchically into clusters; inside a cluster detailed state information is transferred, while among clusters only aggregated information circulates. Private-Network-Network-Interface (PNNI) [7] routing uses such approach, by defining a flexible hierarchical network that can grow up to 104 levels. Scalability gains leads to less optimal paths and complex routing mechanisms.

2.2 Route Calculation

Route calculation can be performed using two main techniques: source routing and distributed algorithms.

In the **Source Routing** approach each node has a global view of the network and routes are calculated at the source using this information, and piggybacked into every data packet. The entailed overhead precludes its use in large scale networks or under heavy load conditions [8].

The **Distributed Routing** attempts to solve this problem by delegating to each node the task of calculating a part of the path toward the destination. Link-state or distance vectors algorithms can be used. Their use in large networks may introduce a significant overhead, leading to the existence of hierarchical solutions, like the one presented earlier for PNNI or even OSPF.

One of the most important problems in route calculation for QoS routing protocols is related to the fact that routes can no longer be defined based on the number of hops. For instance, if the metric is bandwidth, the best route is the one that maximises bandwidth over the bottleneck link, while if the metric is delay, the best route is the one that minimises it; finally, if both metrics are considered, one needs to maximise bandwidth while reducing delay. In most of the cases the problem can be solved by using modified versions of Dijkstra's algorithms.

Another issue that must be considered is the number of paths that are calculated between each pair of source and destination nodes. If a **single path** is used, routing oscillations arise, as long as multi-hop selection is used. This instability problem can be avoided by using load balancing techniques, which can be applied if **multiple paths** are calculated. In [9] it is proposed an algorithm that provides multiple paths of unequal costs to the same destination.

2.3 Path Selection

Today the Internet uses the datagram service model, where paths are selected in a **hop-by-hop** way, using the network's destination address information contained in the packet; most of the existing routing schemes are based on this principle.

Claiming that BE traffic must be routed differently than priority one, new hop-by-hop routing proposals that support service differentiation have recently arised [10] [11]. Nevertheless, as long as the same routing tasks are performed at both edge and core network elements, a significant burden of information processing is spread across the network. In the NGI, complexity must rely on the edge of the network, in order to allow a faster processing at the core, which means that alternative path selection approaches might be more adequate.

As soon as service differentiation becomes an issue, the notion of flow is fundamental to provide QoS support and it might be used to facilitate the cooperation among routing and resource allocation policies, as a virtual service model can be envisaged [15]. By using **Flow Level** routing traffic may be easily routed according to its class of service. In [12], Nahrstedt and Chen propose a combination of routing and scheduling algorithms where priority traffic is deviated from paths congested by BE traffic. Another proposal was made in [13], where QoS traffic uses less congested paths. However, both of them use source routing paradigm, which is not adequate for NGI, as stated before. IETF has proposed a QoS routing framework [14] that performs the flow level path selection; under this proposal every incoming flow is admitted into the network, only if there are enough available resources; otherwise it is blocked. Despite the accuracy that can be achieved with this type of approach, it is very complex and may not scale well, if individual flows are considered.

Scalability may be achieved if instead of using individual the Flow Level routing, an **Aggregated Flow Level** strategy is used to perform path selection. This strategy is complaint with IPv6 standard that provides a Flow Label field in the IP packet header, and may be supported over MPLS networks. Moreover, more complex routing decisions can be rely on the edge of the network and only when traffic flows initiate their activity.

3 Multi-Service Routing

In this section the main characteristics of the **Multi-Service** routing are described. A more detailed description of its architecture can be found in [16]. In this paper a more complete study of the proposed routing protocol will be presented.

3.1 General Principles

The Multi-Service routing proposal extends traditional distributed intra-domain routing protocols, by triggering routing table update cycles, whenever service fulfilment may not be accomplished due to the existing network conditions. Smooth variant quantified metrics are used to trigger such updates, based on global network state information. To assure compatibility, standard mechanisms and messages are used in this updating process.

In spite of using an hop-by-hop approach, an aggregated flow level strategy is used, enabling a scalable and efficient solution. Aggregated traffic flows are defined at the edge of the network by assigning a Flow Label value to the respective field of IPv6's packet header. Complexity relies on the network's edge, as flow identification and maintenance are performed only at the edge routers. Unless re-routing is needed, routing decisions are taken only once, when a new flow is detected; subsequent packets are routed based on their associated aggregated flow service class.

At each time, each router may have two different routing tables: the **standard table**, describing the set of shortest paths to the destination, and the **alternative table**, describing a set of longer paths to the destination. The selection between these tables must be made according to the following set of routing policies:

- Priority traffic should be routed through a standard (shortest) path, as this one has a higher probability of assuring the required service level.
- If the network is less loaded, the remaining traffic may share the same path, as it will not interfere with the performance of higher priority traffic.
- As the network load increases, alternative paths will be found, which will be used by incoming lower priority aggregate flows, in order to meet the level of service of the already active flows and to utilize the unused network resources.
- In case severe local congestion takes place, existing lower priority aggregate flows may need to be re-routed to the alternative path.

3.2 Network State Maintenance

The Multi-Service routing was conceived to avoid complexity. Thus, it uses a Global State strategy and instead of using different measures to evaluate each node's neighbourhood state, a single and simple one was selected: the output link occupancy, which is periodically sampled. Based on the samples an indicator is evaluated using an exponentially weighed moving average (EWMA) technique.

Considering two adjacent nodes i and j and a link $l_{(i,j)}$ connecting them, a number of samples N and a weight α, the output link occupancy indicator, $L_{(i,j)}$, regarding the connection of node i toward node j, at the sampling time t_i is given by:

$$L_{(i,j)}(t_i) = \alpha * \frac{\sum_{t=t_{(i-1)-N}}^{t=t_{i-1}} L_{(i,j)(t)}}{N} + (1-\alpha) * L_{(i,j)}(t_{i-1}) \qquad (1)$$

Threshold values are defined and, in order to avoid nasty traffic balance oscillations effects, a hysteresis mechanisms is also considered. Whenever a threshold is reached, a quantified QoS metric is modified and the alternative routing table update procedure is triggered.

When $M_{(i,j)}(t)$ represents the value of the QoS metric between node i and j at sampling time t; T_k represents the k^{th} threshold; H_k the associated hysteresis value and M_k the corresponding metric. At a sampling time $t_i > t$ link $l_{(i,j)}$ changes its QoS metric, as long one of the two following conditions apply:

$$L_{i,j}(t) < T_k \wedge L_{i,j}(t_i) \geq T_k \Rightarrow M_{(i,j)}(t_i) = M_k \quad (2)$$

$$L_{i,j}(t) \geq T_k \wedge L_{i,j}(t_i) < T_k - H_k \Rightarrow M_{(i,j)}(t_i) = M_0 \quad (3)$$

Two major threshold values were defined:

- **Deflection Threshold** - it acts like a type of pre-congestion alert; when it is reached, all previous traffic flows keep their paths, while the new incoming lower priority traffic flows are routed according to the new alternative routing table's paths that will surely not include the current link.
- **Critical Threshold** - it causes the removal of all low priority traffic flows that are currently crossing the critical link. This removal is done by a signaling mechanism that notifies a set of border routers to take the appropriate actions to reroute their incoming lower priority traffic flows that are crossing the saturated node's link at the time. Border routers determine new paths to those flows by deleting related ones.

Hysteresis is also defined as **Standard Thresholds**. When they are reached, it means that a steady light traffic load condition persists in the node's link and the paths containing this link will be available, again, to the new low priority traffic flows.

3.3 Route Calculation

Multi-Service routing is an extension of traditional intra-domain routing protocols, being able to use a link-state or a distance vector approach. Routing information is distributed to all routers in the domain. If a Link-State routing strategy (OSPF) is used, two independent instances of the routing protocol are executed at each node. One of them periodically transfers Link State Advertisements (LSAs), which carry the administrative metric, and updates the standard routing table, accordingly; the other one uses LSAs to disseminate QoS metric and updates the alternative routing table. In order to have multiple paths per destination, a modified version of the Dijkstra algorithm is used in each routing instance. If a Distance Vector routing protocol (RIP) is used, the same type of structure is employed: two independent instances of the protocol are used, one uses the administrative metric and computes the standard path, while the other uses the QoS metric and computes the alternative path. Multiple paths per destination for each service class leads to the utilisation of a modified version of Bellman-Ford algorithm.

Administrative information is periodically transferred to assure consistency of routing information, but also when a topological change occurs. As regarding QoS information, the network state may change very often, leading to frequent changes in QoS metrics. To avoid a burden of routing traffic due to such situations and routing instabilities, QoS routing information is transferred periodically or when there is a change on a QoS metric that occurs after a stability period since the last change. Thus, very frequent changes are only advertised if they persist after that period of time.

Considering link $l_{(i,j)}$ and the existence of modifications on its QoS metric $M_{(i,j)}$ that occur in two instants of time, instant t_i and instant $t_i+\delta$; considering also a stability period of T; such modification will only generate an alternative routing table update event, $Ev_{(i,j)}$, if the following condition is verified:

$$M_{(i,j)}(t_i+\delta) \neq M_{(i,j)}(t_i) \wedge \delta \geq T \Rightarrow Disseminate(Ev_{(i,j)}) \tag{4}$$

3.4 Path Selection

The Multi-Service routing path selection strategy is based on a Aggregated Flow Level strategy, being completely different from the traditional intra-domain hop-by-hop method.

At the edge of the network, each incoming new flow is classified into an **Aggregated Service Class**, according to its service class, age and ingress and egress nodes. The first packet of each flow that arrives at each node uses the routing tables (standard or alternative) to identify the next hop; subsequent packets of the same flow are associated with it at the edge of the network; their routing will be based on the flow identifier they carry and on the associated routing information, retrieved by this first packet to select the path.

Considering a packet, $pkt_{(i,t)}$, arriving at node i at instant t; the aggregated service classes $ag\_sc_(z)$, where z represents a specific class and *any* a class among the existing ones; the DiffServ service classes $sc_{(p)}$, where p represents the priority of the class ($Prio$ or BE); the network state's conditions, from node's i perspective, $ns_{(i,s)}$, where s represents the network state (low (L), medium (M) or heavy (H) load conditions); the standard routing table, Std_Rt and the alternative routing table, Alt_Rt; and also the selected next hop $hop_{z,x}$, where x is the node's selected egress interface (s via the standard path and a via the alternative one), the routing policies can be defined as follows:

- $if pkt_{(i,t)} \notin ag\_sc_{(any)} \wedge pkt_{(i,t)} \in sc_{(Prio)} \Rightarrow$ $new(ag\_sc_{(z)}, pkt_{(i,t)}) \leftarrow z_1; select(Std\_Rt_{(i,t)}, pkt_{(i,t)}) \leftarrow hop_{(z_1,x_s)}$
- $if pkt_{(i,t)} \notin ag\_sc_{(any)} \wedge pkt_{(i,t)} \in sc_{(BE)} \wedge ns_{(i,L)} \Rightarrow$ $new(ag\_sc_{(z)}, pkt_{(i,t)}) \leftarrow z_2; select(Std\_Rt_{(i,t)}, pkt_{(i,t)}) \leftarrow hop_{(z_2,x_s)}$
- $if pkt_{(i,t)} \notin ag\_sc_{(any)} \wedge pkt_{(i,t)} \in sc_{(BE)} \wedge ns_{(i,M)} \Rightarrow$ $new(ag\_sc_{(z)}, pkt_{(i,t)}) \leftarrow z_3; select(Alt\_Rt_{(i,t)}, pkt_{(i,t)}) \leftarrow hop_{(z_3,x_a)}$
- $if pkt_{(i,t)} \notin ag\_sc_{(any)} \wedge pkt_{(i,t)} \in sc_{(BE)} \wedge ns_{(i,H)} \Rightarrow$ $new(ag\_sc_{(z)}, pkt_{(i,t)}) \leftarrow z_4; select(Alt\_Rt_{(i,t)}, pkt_{(i,t)}) \leftarrow hop_{(z_4,x_a)}; reroute\ (ag\_sc_{(z4,x_a)})$
- $if pkt_{(i,t)} \in ag\_sc_{(z_i)} \Rightarrow select(hop_{(z_i,x)})$.

4 Simulation Studies

4.1 Simulation Scenario

The proposed routing architecture has been tested through simulations, using the Network Simulator (NS), version 2.27, which has been enhanced with additional capabilities, needed to support this new proposal.Simulations with different network load conditions were performed, using the network scenario described in figure 1 and in table 1.

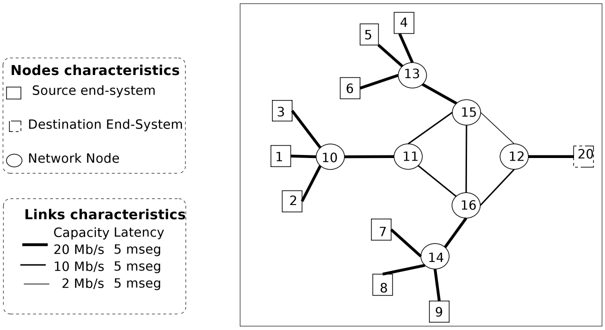


Fig. 1. Network Topology

Table 1. Traffic Characterisation

Class	Type	Number	CoS	Traffic	Src	Dst	Rate[Kb/s]	Size[B]	Total BW [Kb/s]
Prio	Single	1	EF	CBR	1	20	24	40	24
Prio	Aggregate	42	EF	CBR	4	20	24	40	1000
Non-Prio	Aggregate	[0..18]	BE	CBR	5	20	500	1500	[0..9000]

4.2 Parameterisation of Threshold Values

A set of simulations were carried out to configure the thresholds of the Multi-Service routing protocol, in order to adjust the performance of the Multi-Service routing protocol.

In the first set of simulations the Multi-Service routing supports only the critical threshold, which means that when it is reached the entire set of non-priority flows are deviated from the shortest path. This kind of situations should happen only when the network is heavy loaded and thus the threshold values tested are high (80% and 90% of the link occupancy). The threshold that offers the best performance is the one that reduces the losses and delay. As stated in figure 2, although similar results are achieved by both threshold values, fixing the critical threshold at 80% removes the transitory spikes that happened before the path transition occurs and decreases the number of losses in non-priority traffic, which means that a more efficient network utilisation is achieved.

Should the critical threshold be fixed at 80%, the deflection one may be tuned. Three different values were tested (20%, 50% and 70%) and the results are shown

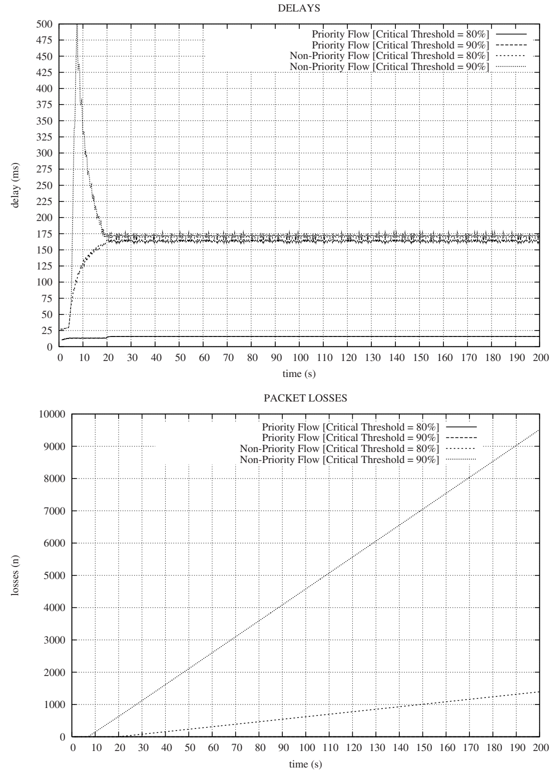


Fig. 2. Critical Threshold parameterisation

on figure 3. If the deflection threshold is adjusted to 20% of the link capacity, incoming non-priority flows starts to be diverted too soon and longer delays are achieved for both priority and non-priority traffic. On the other hand, if the 70% value was selected BE losses will be more significant than those achieved when the deflection threshold is defined at 50%, because the modification of the paths happens too late, when the smaller capacity link (15-12) is already heavy loaded. At 50% of link capacity, both priority and non-priority traffic have a good performance, as delay is kept small and no losses occur in BE traffic.

4.3 Performance Evaluation of Multi-Service Routing

The performance of Multi-Service routing (MS-R) was compared to the performance offered by both the traditional link-state (LS-R) and the QoS routing (QoS-R). The results shown in table 2 illustrates the performance of those algorithms.

For both types of traffic, the Multi-Service routing is the one that presents smaller delays; throughput and losses are similar to those achieved by QoS routing, which are much better than the ones achieved by traditional link-state routing.

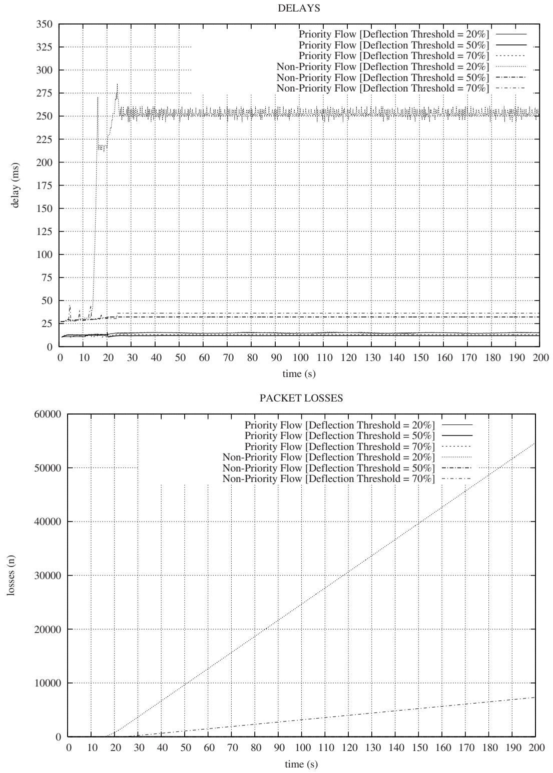


Fig. 3. Deflection Threshold parameterisation

Table 2. Priority traffic: performance evaluation

Priority traffic									
Link load	1 Mb/s			5Mb/s			10 Mb/s		
Type of routing	MS-R	QoS-R	LS-R	MS-R	QoS-R	LS-R	MS-R	QoS-R	LS-R
Delay[ms]	11.91	11.91	11.91	11.95	15.41	10807	11.98	19.9	25813.4
Losses[%]	0	0	0	0	0	80.16	0	0	95.60
Throughput[Mb/s]	0.99	0.99	0.99	0.96	0.96	0.18	0.91	0.91	0.04
Non priority traffic									
Link load	1 Mb/s			5Mb/s			10 Mb/s		
Type of routing	MS-R	QoS-R	LS-R	MS-R	QoS-R	LS-R	MS-R	QoS-R	LS-R
Delay[ms]	-	-	-	28.71	29.06	10807	31.96	30.25	25895.3
Losses[%]	-	-	-	0	0	80.16	0	0	76.34
Throughput[Mb/s]	-	-	-	3.92	3.42	1.78	8.60	8.60	1.93

A more accurate view of the different behaviour of the Multi-Service and the QoS routing protocols is depicted in figure 4.

As can be stated, the Multi-Service routing also presents a more stable longterm behaviour, as no significant traffic spikes occurs. At time instant 3, the deflection threshold is crossed because the output link of node 12 towards node 15 reaches 50% of its capacity; non-priority traffic presents a slightly better

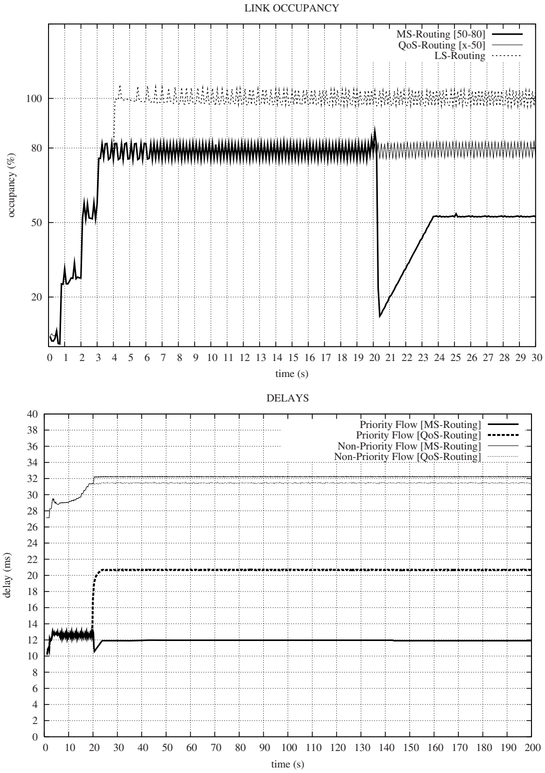


Fig. 4. Longterm behaviour

performance than the one it has presented before, as new incoming non-priority flows are diverted through a longer path. As new priority traffic are still being applied to the network after that time instant, the link occupancy (15-12) stays near 80%, but only at time instant 29, it crosses the critical threshold. At this time, all the non-priority traffic is diverted to a longer path and so the critical link occupancy and the delay of priority traffic sharply decreases. If QoS routing is used, when the threshold is crossed every incoming new flow (priority or non-priority) is transmitted through a longer path. Thus, link occupancy is kept near 80% and the delay of priority traffic increases approximately 80%.

5 Conclusions

Existing QoS routing protocols are not able to deal efficiently with service differentiation. The proposed routing protocol provides this kind of support. To perform this, several extensions which provide a solution compatible with traditional routing protocols, with scalability characteristics, have been proposed. Simulation results have shown that priority traffic will achieve better performance and non-priority traffic will suffer less losses. Future work comprises testing the Multi-Service routing in more complex networks; study of other metrics and the integration into an IPv6/MPLS trial platform.

References

[1] X. Xipeng and M. N. Lionel: Internet QoS: A Big Picture. IEEE Network, March/April (1999).

[2] S. Chen and K. Nahrstedt: An Overview of Next-Generation High-Speed Networks: Problems and Solutions. IEEE Network, November/December (1998).

[3] M. Garey and D. Johnson: Computers and Intractability: A Guide to the Theory of NP-completeness. New York: W. H. Freeman ZhuPar95andCo (1979).

[4] S. Nclakuditi, Z. Zhang and C. H. Du David: On selection of candidate paths for proportional routing. Computer Networks 44 (2004) 79-102.

[5] A. Khanna and J. Zinky.: The revised ARPANET Routing Metric. Proceedings of SIGCOMM'89, September (1989).

[6] R. Guérin, S. Kamat, A. Orda, T. Prygienda and D. Williams: QoS Routing Mechanisms and OSPF extensions IETF RFC 2676, August (1999).

[7] ATM Forum: Private network network interface , Specification Version 1 (PNNI 1.0). March (1996).

[8] R. Guérin and A. Orda: QoS Based Routing in Networks with Inaccurate Information: Theory and Algorithms Proceedings of IEEE Infocom'97, Japan (1997).

[9] S. Vutukury and Garvia-Luna-Acheves: A Simple Approximation to Minimum Delay Bridge. Proceedings of ACM SIGCOMM'99, August/September (1999).

[10] M. Oliveira, B.Melo, G. Quadros and E. Monteiro: Quality of Service Routing in the Differentiated Services Framework Proceedings of SPIE's International Symposium on Voice, Video and Data Communications (Internet III: Quality of Service and Future Directions), Boston, Massachutetts, USA, November 5-8 (2000).

[11] J. Wang and K. Nahrsted: Hop-by-hop Routing Algorithms for Premium-Class Traffic in DiffServ Networks. Proceedings of IEEE INFOCOM 2002, New York, June (2002).

[12] K. Nahrstedt and S. Chen: Coexistence of QoS and Best Effort Flows - Routing and Scheduling Proceedings of the 10th IEEE International Workshop on Digital Communications: Multimedia Communications, Ischia, Italy, September (1998).

[13] Q. Ma and P. Steenkiste: Support Dynamic Inter-Class Resource Sharing: A Multi-Class QoS Routing Algorithm Proceedings of IEEE INFOCOM'99, New York, March (1999).

[14] E. Crawley, R. Nair, B. Tajagopalan and H. Sandick: A Framework for QoS based routing. IETF RFC 2386, August 1998.

[15] L. Cidon, R. Rom and Y. Shavitt: Multi-path Routing Combined with Resource Reservation. Proceedings of IEEE INFOCOM'97, Japan, April (1997).

[16] A. Varela, T. Vazão and G. Arroz: Multi-Service Routing: a New QoS Routing Approach Supporting Service Differentiation. Proceedings of AICT'05, Lisbon, April (2005).

Quantifying the BGP Routes Diversity Inside a Tier-1 Network

Steve Uhlig[⋆] and Sébastien Tandel

Department of Computing Science and Engineering,
Université catholique de Louvain, Louvain-la-neuve, B-1348, Belgium
{suh, sta}@info.ucl.ac.be

Abstract. Many large ISP networks today rely on route-reflection [1] to allow their iBGP to scale. Route-reflection was officially introduced to limit the number of iBGP sessions, compared to the $\frac{n\times(n-1)}{2}$ sessions required by an iBGP full-mesh. Besides its impact on the number of iBGP sessions, route-reflection has consequences on the diversity of the routes known to the routers inside an AS. In this paper, we quantify the diversity of the BGP routes inside a tier-1 network.

Our analysis shows that the use of route-reflection leads to a very poor route diversity compared to an iBGP full-mesh. Most routers inside a tier-1 network know only a single external route in eBGP origin. We identify two causes for this lack of diversity. First, some routes are never selected as best by any router inside the network, but are known only to some border routers. Second, among the routes that are selected as best by at least one other router, a few are selected as best by a majority of the routers, preventing the propagation of many routes inside the AS. We show that the main reason for this diversity loss is how BGP chooses the best routes among those available inside the AS.

Keywords: BGP, iBGP, route-reflection, route diversity.

1 Introduction

The Internet consists of a collection of more than 21,000 domains called Autonomous Systems (ASs). Each AS is composed of multiple networks operated under the same authority. Inside a single domain, an independent Interior Gateway Protocol (IGP) [2] such as IS-IS or OSPF is used to propagate routing information. Between ASs, an Exterior Gateway Protocol (EGP) is used to exchange reachability information. Today, BGP [2] is the de facto standard interdomain routing protocol used in the Internet. BGP routers exchange routing information over BGP sessions. External BGP (eBGP) sessions are established over inter-domain links, i.e., links between two different ASes (BGP peers), while internal BGP (iBGP) sessions are established between the routers within an AS.

Route-reflection [1] was initially introduced as an alternative to the iBGP full-mesh that requires $\frac{n\times(n-1)}{2}$ iBGP sessions to be established inside an AS. This number of sessions required for propagating the routes learned from the neighbors of the AS to all routers inside the AS does not scale for large networks containing hundreds or thousands of BGP routers. Route-reflection [1] was thus introduced to limit the number of

[⋆] Steve Uhlig is Postdoctoral fellow of the Belgian National Fund for Scientific Research (F.N.R.S).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1002–1013, 2006.

iBGP sessions for large sized networks. An advantage of an iBGP full-mesh is that all routers know about all the best routes of the other routers inside the network. This means that when some route is withdrawn, routers can typically switch to another route immediately, without waiting for BGP to converge. Without the use of an iBGP full-mesh on the other hand, routers might know only a single route to reach an external destination. When this route is withdrawn, then the concerned prefix will not be reachable until BGP reconverges and advertises an alternative route. BGP is known to suffer from slow convergence [3]. BGP routes diversity is thus important to understand if high availability of the reachability service is to be provided, as is typically the case in tier-1 providers.

Route-reflection inside an AS defines two types of relationships among BGP routers: client and non-client. These relationships among BGP peers define a loose hierarchy among routers, going from the bottom level routers that have no clients up to the largest route-reflectors that are not client of any other router. Note that this implicit hierarchy is not practically enforced, as iBGP sessions can established between any two routers inside the AS, even under route-reflection.

The redistribution of the routes in BGP works according to well-defined rules. First recall that a route is never re-advertized to the peer that announced it. Consider a given prefix p for which a router inside the iBGP receives several routes from its peers (iBGP or eBGP). The router chooses among the possible ones towards p its best route using the BGP decision process [4].

How the best route is propagated to the neighbors of a router depend on whether the router acts as a router-reflector. If a router does not act as a route-reflector, i.e. it has no "client" peer, then the router advertises this route to all its iBGP peers if it is learned from an eBGP session, or to none of them if the route was learned from an iBGP session. If a router acts as a route-reflector [1] on the other hand:

- If the route was learned from a client peer (or eBGP peer), the route-reflector redistributes the route to all its clients and non-client peers (except the one from which the route was received).
- If the route was learned from a non-client peer, the route-reflector redistributes the route to its client peers only.

These rules driving the redistribution of the routes inside the iBGP imply an implicit filtering of the routes over the internal BGP signaling graph. Besides the rules defined in [1] when connecting route-reflectors to ensure a proper working of the iBGP propagation, there is no clear design rules known today as to how to design a proper iBGP graph. Guidelines for checking that a correct iBGP configuration have been discussed in [5]. [6] provides a tool to detect potential problems due to the iBGP configuration based on static analysis.

Route-reflection was initially proposed as an alternative to the full-mesh, but in practice it caused many problems and it is unclear what it actually performs on which routes are propagated compared to a full-mesh. In this paper, we thus aim at quantifying the diversity inside a tier-1 network that relies on route-reflection. We see our work as a first step towards a better understanding of the impact of route-reflection on route diversity.

2 Methodology

Unless one has complete data concerning the full topology, the configuration of the routers, and the eBGP routes learned by an AS, it is not possible to correctly reproduce its routing state [7]. This is the main reason why typically, simulations have to be used to reproduce the routing of a large AS. The aim of this section is to sketch our methodology to reproduce the routing of the studied network.

We relied on CBGP [7] to model our tier-1 network. For this, we used the physical topology of the network (links and IGP weights), as well as the configuration of the BGP routers. We obtained the Adj-RIB-In's from the main route-reflectors of the studied network. Because the BGP routes present in the Adj-RIB-In's of internal routers do not always contain the information about which eBGP peer actually originated a route, some reverse-engineering of the route origin was necessary. We could of course keep the routes learned directly from eBGP sessions, as large route-reflectors also have a significant number of eBGP peerings (see Section 5). Two cases are possible when trying to find the entry point of a route:

- The BGP next-hop of the route is the IP address of an external peer. In this case we must pay attention to advertise this route from the external peer found to the internal router with which the external peer has established the eBGP session.
- The BGP next-hop of the route is the IP address of an internal router because this router has been configured with *next-hop-self*. We have to find the originating external peer that advertised the route to the internal router. To find it, we rely on the AS path information. We search for eBGP peers belonging to the leftmost AS on the AS path that have an eBGP peering with the internal router.

To ensure that our model was correct we validated the conversion by injecting the routes in the model and then checked the routes computed by the model against the original best routes seen in the route-reflectors. Due to space limitations, we do not provide these results here.

As even in our simulation model, it is not always possible to identify the eBGP peer from which a route has been advertized by looking at the route, all external routes in the C-BGP simulation had to be tagged with a special community value identifying the external router from which the route was learned. This made the analysis of the results easier. Once all external routes were identified, C-BGP [8] performed the propagation of the routes according to the internal iBGP structure of the network, and we retrieved the content of the Adj-RIB-In's of all routers inside the C-BGP simulation. Our analysis is based on the outcome of this simulation.

Among all prefixes of our input data, we selected a subset of them (940). Those 940 prefixes were learned from several locations in the network. In the analysis of this paper, only those 940 multiply-advertized prefixes are considered as measuring diversity for singly-advertized prefixes is meaningless. We selected the largest of them in terms of the amount of traffic sent towards them. These prefixes captured 80% of the total traffic according to the Netflow [9] statistics. 80,000 destination prefixes were present in the Netflow statistics, most of them representing an insignificant fraction of the total traffic.

The iBGP structure of the studied network consists of 3 levels of route-reflection according to which router is a client of which other router. This graph contains 105 nodes (routers) partitioned into 36 geographically distinct POPs and 169 undirected edges. This iBGP "hierarchy" is a static one, by design of the iBGP graph. To find out the hierarchy inside the route-reflection graph, we rely on a topological sort of a directed acyclic graph (DAG) [10]. The reason why we have to rely on this concept of a DAG is that the route-reflection graph is not a strict hierarchy (a forrest). Contrary to a general misbelief, route-reflection does not require a strict hierarchy to work. A strict hierarchy is even not desirable for route diversity. The vertices of the route-reflection graph (rr_graph) are all routers inside the iBGP graph. An arc (i,j) of the route-reflection graph rr_graph connects a reflector (i) to a client router (j). The *level* in the route-reflection hierarchy is computed by finding out which reflectors are not clients of any other router in the reflection graph. These are given a level of 0 in the hierarchy, they are the top-level route-reflectors of the graph (16 routers). Route-reflectors which are clients of the top-level (0) reflectors have a route-reflection level of 1 (57 routers). Finally, clients of reflectors at level 1 are given a level of 2 (32 routers).

3 Example of Route Diversity Loss

When relying on an iBGP full-mesh, all the external routes selected as best by the border routers are known to all other routers inside the AS. An iBGP full-mesh is thus "ideal" in terms of the diversity of the routes known to all routers inside an AS, at the cost of a large number of iBGP sessions. Even this "ideal" situation might hide some eBGP routes when a border router has multiple eBGP sessions or when it does not choose as its best route one among its eBGP-learned ones. This would happen if one of its non eBGP-learned routes has a higher local-pref or smaller AS path length than its eBGP-learned routes. A loss in diversity will thus occur only because of this order of the rules of the BGP decision process.

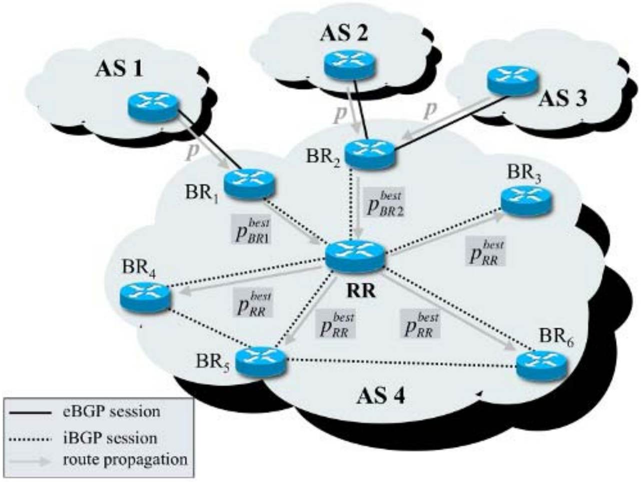


Fig. 1. Example of route loss inside iBGP

For instance, Figure 1 illustrates the two main causes for loss of diversity on an example. Prefix p is advertized to AS4 by 3 neighboring ASes (AS1, AS2 and AS3), two of them at border router $BR2$ and another at border router $BR1$. eBGP sessions are indicated by solid lines, while iBGP sessions by dashed lines. Arrows indicate the propagate of a route from one router to another. Only the best route chosen by $BR2$, let us call it p^{best}_{BR2}, will be propagated inside AS4. The best route propagated by $BR1$, assuming it is the external one (p^{best}_{BR1}), will also be propagated within AS4. Route reflector RR is on the iBGP propagation path of both routes p^{best}_{BR1} and p^{best}_{BR2}, hence it will choose at most one of these as best route, which we call p^{best}_{RR}. As we have one route reflector in AS4, all other routers are clients, hence because of the iBGP propagation rules RR will redistribute its best route to all its clients except the one from which it learned the route.

To prevent this loss of diversity, several solutions can be envisioned. First, one can change the location of the eBGP peerings so as to minimize the loss of the routes at the border routers. Changing the location of eBGP peerings is typically not practically feasible because it depends on the slots available on the routers and the geographical constraints about where peers can connect to the routers of the AS. Another solution is to reconfigure the iBGP graph by adding and removing iBGP peerings between routers, but this operation is tricky as it is difficult to predict its impact on the BGP propagation [5, 6]. Finally, redistributing more than a single route [11] could be seen as a solution. This would however require changes to the protocol at the risk of creating divergence. A proper understanding of route diversity is thus necessary before thinking about changed in how routes are propagated inside an AS.

To show to what extent external routes can be lost only due to multiple eBGP peerings at the same border router, Figure 2 compares for each prefix the total number of known external routes with the number of routes that will never be selected as best due to multiple peerings at the edge routers, in the studied tier-1 network. Each border router may receive several external routes from its eBGP peers for a given prefix. The points labeled "lost" sums for each prefix (over all border routers) the number of external routes that cannot be chosen as best because several are received by a border router.

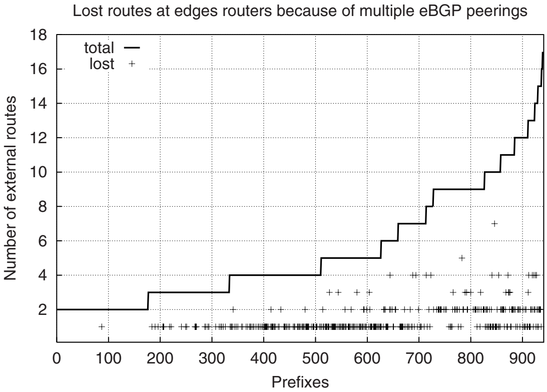


Fig. 2. External routes lost at edge routers

534 over 5018 routes are lost because of multiple external routes received by border routers. Hence more than 10% of the external routes cannot be considered just because of the location of the eBGP peerings inside the network. These lost routes concern 365 over the 940 prefixes, 40% of the considered prefixes for which several external routes are known.

4 iBGP Structure of the Studied Network

In this section, we want to highlight two points. First, we want to make clear that the hierarchy induced by route-reflection and the propagation of the routes inside the AS are two very different things. Second, we want to discuss how much the location of a router inside the iBGP propagation graph varies across prefixes.

The propagation inside iBGP depends on from which border routers the routes were learned. Each prefix can be learned from a different set of border routers, even though most of the prefixes are typically learned from a small subset of all possible border routers. In the studied network, eBGP peerings can be attached to any router, from level-0 reflectors to routers at the edge of the network (level-2). Centrality in the reflection hierarchy hence does not match the centrality of a router inside the iBGP propagation graph.

Directly comparing the *level* of a router with its location in the signaling graph is problematic for two reasons. First, the *level* of a router is a very discrete variable taking only 3 different values. Second, the variation of the location of a router from the eBGP peering wherefrom the route has actually been learned by the AS varies a lot. We define the depth $depth(r, p)$ of a router r in the iBGP signaling graph for a given prefix p as the number of iBGP hops it took for the best route chosen by r towards p from the eBGP peer who advertized this route. The $depth(r, p)$ varies between 1 and 6 in our studied network. Still, the typical values of the depth lies around 2 and 3 for most routers. The routers having many eBGP peerings or that are central (level-0 reflectors) inside the iBGP graph tend to have a smaller depth than less central routers (level-2 reflectors).

5 Best Route Choice and Route Origin

An important factor to understand the propagation of the routes inside the iBGP is from what kind of peering the best routes of a router were learned by any router. Figure 3 provides the breakdown of the best routes chosen by each router according to what type of BGP peer advertized the route. A route can be learned either from an eBGP peer, a client peer (for route reflectors) and a non-client peer (both for reflectors and other routers). Routers on the x-axis of Figure 3 are ordered by their increasing level inside the route-reflection hierarchy, so the first 16 routers are level-0 reflectors, the next 57 level-1, and the last 32 are level-2 reflectors. This ordering of the x-axis was chosen because one might expect that more central routers like level-0 reflectors would have a larger fraction of their best routes learned from the iBGP. The y-axis of Figure 3 gives, for each router, the percentage of best routes of each type. For each router, we computed among the best routes it selected, the fraction of them that have been learned from eBGP sessions, client and non-client sessions. On Figure 3 we plot the fraction

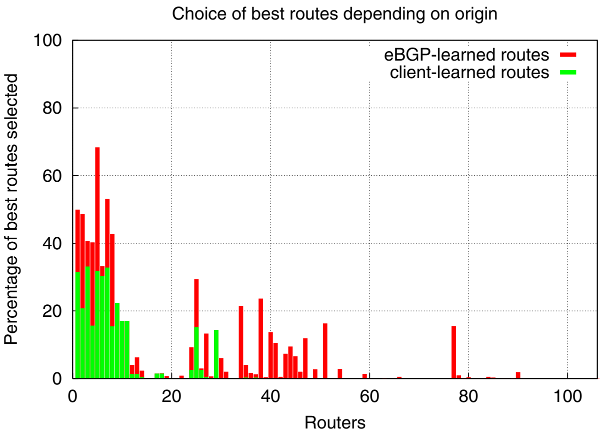


Fig. 3. Breakdown of best route choice by origin

of client-learned routes, then the sum of client-learned routes and eBGP-learned ones. Non-client-learned routes are not shown on Figure 3 but make the rest of the 100% of the best routes.

It is easy to see that excepted for level-0 reflectors (the first 16 routers), most routes are non-client routes, i.e. routes learned from either a reflector from which the local router is a client or a regular iBGP peer. Only large reflectors (mainly level-0) select routes learned by client peers, as these routers also have the largest number of client peers. Note that routers for which it might seem on Figure 3 to have only selected as best non-client-learned ones actually have typically a few eBGP-learned or client-learned routes as best. This is not apparent from the use of the percentage over all considered prefixes.

Figure 3 told that most best routes are learned from iBGP peers. However, this choice of the best routes might be biased by a lack of eBGP peerings at some routers. This is however not the case in the studied network, as non-client peerings represent 53% of the total peerings, client peerings about 23%, and eBGP peerings about 26%. More than one fourth of all BGP sessions are thus eBGP sessions, hence a lack of eBGP peerings is not the reason why routers do not select their best route from a eBGP peer-learned one.

95% of the best routes are learned from non-client peers, about 2% from client peers, and 3% from eBGP peers. Most routes chosen as best by the routers come either from a regular iBGP peer or a route-reflector of which the considered router is a client. The choice of the best route of a router thus depends a lot on the choice performed by the route-reflectors higher in the hierarchy. This phenomenon is caused by the relatively small number of locations from which a prefix is learned by the AS, hence the iBGP propagation graph is very important to understand which route will be propagated inside iBGP.

6 Measuring iBGP Route Diversity

To measure the diversity of the routes, we define two metrics: the *real diversity* and the *RIB diversity*. The choice of these metrics mainly reflects our own interest of understanding what fraction of the external routes is actually known to the routers inside the iBGP compared to those know to the whole AS. Let us insist on the fact that as the route-reflection graph is not a forest, a given eBGP-learned route can be propagated through different iBGP propagation paths. It thus makes sense to measure the difference between the number of actually distinct routes a router learns from its neighbors and how this number relates to from how many distinct eBGP peers those routes come.

The *real diversity* measures the proportion of the external routes known by the AS any router has learned. The *real diversity* $div_{real}(r, p)$ counts for each router r and prefix p the number of unique external routes (learned from distinct eBGP peers) r has in its Adj-RIB-In's divided by the total number of eBGP routes that have been learned by routers of the AS:

$$div_{real}(r, p) = \frac{routes_{unique}(r, p)}{routes(p)}. \tag{1}$$

$routes(p)$ denotes the total number of distinct eBGP routes (learned from different eBGP peers) known by all routers of the AS and $routes_{unique}(r, p)$ the number of distinct eBGP routes r has in its Adj-RIB-In's for prefix p. Even in an iBGP full-mesh, some routers will not forcibly have a div_{real} of 1 when they learn multiple eBGP routes since they can propagate only a single route inside the iBGP.

The *RIB diversity* $div_{RIB}(r, p)$ on the other hand counts for each router r and prefix p the number of unique external routes (learned from distinct eBGP peers) r has in its Adj-RIB-In's divided by the total number of entries in its Adj-RIB-In's:

$$div_{rib}(r, p) = \frac{routes_{unique}(r, p)}{rib(r, p)}. \tag{2}$$

$rib(r, p)$ denotes the number of Adj-RIB-In entries router r has for prefix p. $div_{rib}(r, p)$ takes values in the]0,1] range. If r has no route towards p then its *RIB diversity* will be undefined. The closer to 1 the value of div_{rib}, the less redundancy there is among the routes r knows towards p.

In practice, one would like as high a value of both metrics. If many external routes are known inside the AS, then the value of div_{real} will be low so that a low value of div_{real} is not an indication of a "bad" diversity. A value of div_{rib} smaller than 1 indicates that among the several routes a router learns, some of them are duplicates and will thus be withdrawn if the corresponding external route is withdrawn. Such redundant iBGP routes protect a router from the failure of one of the routers that advertise this route.

7 Real and RIB Diversity of the Studied Network

On Figure 4, we show for each considered prefix p the average over all routers of the network of $div_{real}(., p)$, $div_{rib}(., p)$, and $\frac{1}{routes(p)}$. Prefixes on the x-axis of Figure 4 are ordered by increasing value of $routes(p)$. Recall that $routes(p)$ denotes the number of different eBGP peers from which a route towards p is learned. The reason for plotting $\frac{1}{routes(p)}$ on Figure 4 is that it provides a lower bound on $div_{real}(., p)$, i.e. it is the value of $div_{real}(., p)$ if routers only know no more than a single unique external route towards p.

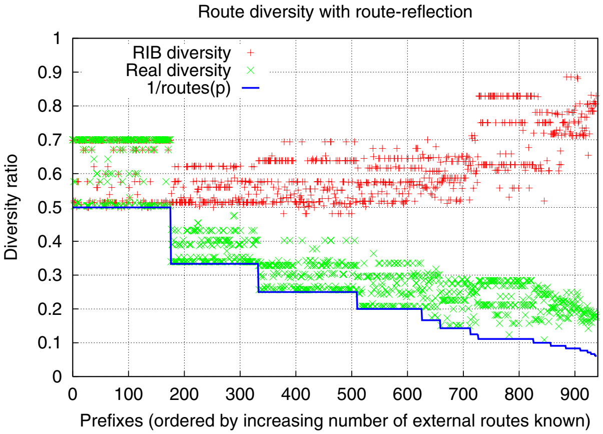


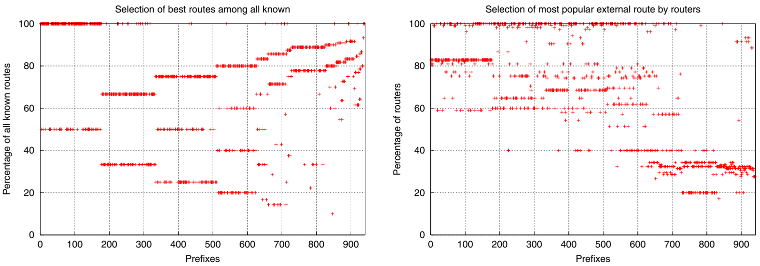
Fig. 4. External diversity for each prefix

The main message from Figure 4 is how closely the *real diversity* curve follows the inverse of the number of total eBGP routes known to the AS. On average, routers know not much more than a single unique route (in terms of its eBGP origin) for any given prefix. This observation implies that the current iBGP structure of the studied network does not provide diversity in terms of the external routes. Furthermore, the value of the *RIB diversity* is about 0.5 for a large fraction of the prefixes. About half the entries in the Adj-RIB-In's are duplicate routes in terms of the eBGP peer who advertized the route inside the AS. This reflects the design choice of the studied network, which connects routers to several iBGP peers but the latters advertise the same eBGP-originated route. Note that as we ordered prefixes by increasing number of eBGP routes known, the large variations in this number of eBGP routes known for prefixes (up to 17) is pretty important, see the "$\frac{1}{routes(p)}$" curve.

8 Route Sampling Performed by BGP Route Selection

Which routes are chosen as best by the routers inside an AS are another important factor that explain diversity inside the iBGP. Among all the routes advertized by eBGP peers, a subset of them are preferred by BGP routers because of the BGP decision process chooses the best route. From which kind of neighboring AS the route comes, its AS path length, and other attributes of the routes are a key factor for determining which routes will never be selected as best by any router inside the AS.

Figure 5(a) provides for each prefix the percentage of all external routes known that have been selected as best route by at least one router inside the network. The prefixes on both graphs of Figure 5 (x-axis) have been ordered by increasing number of external routes known ($route(p)$). There are three regions on Figure 5(a) that correspond to three different types of prefixes. The first type are those prefixes for which all external routes have been selected as best by at least one router inside the AS. For these prefixes, no external route is lost at the border routers. Note that most of the prefixes for which there is no loss of external routes at border routers are mainly those having only 2 external



(a) Proportion of routes selected as best (b) Proportion of routers that select most widely selected route

Fig. 5. Selection of best routes among all known

routes known. It is very unusual that prefixes having a higher number of external routes have all their external routes selected by at least one router.

The second type of prefixes are those for which only a single route is selected by all routers. 192 over the 940 prefixes having more than 2 external routes known inside the AS have only a single external route selected by all routers of the AS. The reason why these prefixes have only one route chosen as best by all routers is that routers prefer this single route over the others.

Finally, the third type of prefixes are in-between, with some routes lost at border routers, but more than one route is selected as best by at least one router. We can also observe on Figure 5(a) that prefixes for which a large number of external routes are known tend to have a large fraction of these external routes selected as best by at least one router.

Figure 5(b) gives for each prefix, the fraction of all routers that selected the most popular among all the known routes. By *most popular route*, we mean the route which was selected as best by the largest number of routers inside the AS. Among the subset of the routes that are selected as best by at least one router, the one that is selected as best by the largest number of routers is chosen by a very large fraction of the routers compared to other routes. Obviously, all prefixes of the second type according to the previous paragraph will appear as points with 100% of the routers having selected the same route on Figure 5(b). We can see on Figure 5(b) that most points lie above 50%, except for prefixes having a very large number of external routes known. For the latters, the choice of the best route is less biased towards a single route.

9 Route Diversity Per Router

Even though the previous section showed that the choice of the best routes inside the studied network favors a loss in route diversity across the iBGP graph, we would expect that diversity is still present somewhere in the AS. We might expect that the iBGP signaling graph under route-reflection limits the number of iBGP sessions compared to a full-mesh, but without removing all the route diversity known across the whole AS. In this section, we want to see whether there are differences among routers in terms of route diversity. A desirable goal would be that all routers know two unique routes for

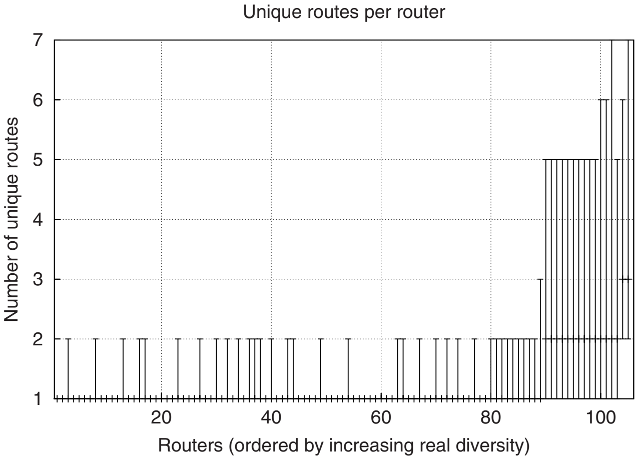


Fig. 6. Unique routes known to routers

each prefix. In such a case, even if the current best route is withdrawn the router can switch immediately to the alternative route. Note that if the route is withdrawn due to a failure inside the AS or at the peering link over which the route was announced then local protection can be used.

Figure 6 show, for each router, how many unique external routes it knows towards any prefix. The y-axis of Figure 6 gives the 20 and 80 percentiles of this number of unique routes for each router over all considered prefixes. The ends of the bars show the 20 and 80 percentiles. Routers on the x-axis of Figure 6 are ordered by increasing median of their *real diversity*.

Figure 6 shows that some routers (the rightmost ones) have a large number of unique routes in their Adj-RIB-ins for most prefixes. These routers having diversity are both level-1 and level-0 reflectors. However, many routers have a value of 1 both as their 20 and 80 percentile. This means that these routers only know 1 unique external route for most of the prefixes. These are mainly level-2 routers in the route-reflection hierarchy. This is something to be expected in a real network as most clients are topologically close to their route-reflectors, hence even though they might be connected to several higher level route-reflector, they will receive the same route (in eBGP origin) from the route-reflectors they are peering with. The iBGP structure of the studied network hence does not lack diversity, but diversity is very unevenly distributed among the routers. A few routers (top-level route-reflectors) have a very high diversity while most routers know only a single route.

10 Conclusion

In this paper, we quantified the diversity of the routes inside a tier-1 ISP. By building a model of the tier-1 ISP and reproducing its routing, we tried to better understand how its iBGP structure impacts its BGP route diversity.

We showed that the impact of the use of route-reflection on route diversity is significant. Most routers of our tier-1 network typically only know a single external route towards a destination prefix. Its iBGP graph propagated redundant routes that are not externally distinct from eBGP origin.

We identified two causes for this lack of diversity. First, some routes are never selected as best by any router inside the network, but are known only to one border router. Second, among the routes that are selected as best by at least one router, a few are selected as best by a majority of the routers, preventing diverse routes to propagate across the AS.

Our results point to the big distance in terms of route diversity between route-reflection and an iBGP full-mesh. Route-reflection thus reduces the number of iBGP sessions at a high cost in limiting the diversity of the routes inside the AS. Routes diversity inside an AS is important in case of failures, to ensure that all routers always have a route during the convergence of BGP after a failure. Our work hence calls for a deeper understanding of the possible trade-offs between iBGP route diversity, scalability and safety in the convergence of BGP.

Acknowledgments

We would like to thank Olaf Maennel for many insightful suggestions about the presentation of this paper.

References

1. T. Bates, R. Chandra, and E. Chen, "BGP Route Reflection - An Alternative to Full Me sh IBGP," Internet Engineering Task Force, RFC2796, April 2000.
2. B. Halabi and D. Mc Pherson, *Internet Routing Architectures (2nd Edition)*, Cisco Press, January 2000.
3. C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "An experimental study of Internet routing convergence," in *Proc. of ACM SIGCOMM*, August 2000.
4. Cisco, "BGP best path selection algorithm," http://www.cisco.com/warp/public/459/25.shtml.
5. Timothy G. Griffin and Gordon Wilfong, "On the correctness of iBGP configuration," in *Proc. of ACM SIGCOMM*, August 2002.
6. N. Feamster and H. Balakrishnan, "Detecting BGP Configuration Faults with Static Analysis," in *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, May 2005.
7. B. Quoitin and S. Uhlig, "Modeling the routing of an Autonomous System with C-BGP," *IEEE Network Magazine*, vol. 19, no. 6, November 2005.
8. B. Quoitin, "C-BGP, an efficient BGP simulator," http://cbgp.info.ucl.ac.be/, September 2003.
9. Cisco, "NetFlow services and applications," White paper, available from http://www.cisco.com/warp/public/732/netflow, 1999.
10. T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*, MIT Press and McGraw-Hill, 2001.
11. D. Walton, A. Retana, and E. Chen, "Advertisement of Multiple Paths in BGP," Internet draft, draft-walton-bgp-add-paths-04.txt, work in progress, September 2005.

Distributed QoS Routing for Backbone Overlay Networks

Li Lao[1], Swapna S. Gokhale[2], and Jun-Hong Cui[2]

[1] Computer Science Dept., University of California, Los Angeles, CA 90095
[2] Computer Science & Engineering Dept., University of Connecticut, CT 06029
llao@cs.ucla.edu, {ssg, jcui}@engr.uconn.edu

Abstract. In recent years, overlay networks have emerged as an attractive alternative for supporting value-added services. Due to the difficulty of supporting end-to-end QoS purely in end-user overlays, backbone overlays for QoS support have been proposed. In this paper, we describe a backbone QoS overlay network architecture for scalable, efficient and practical QoS support. In this architecture, we advocate the notion of QoS overlay network (referred to as QSON) as the backbone service domain. The design of QSON relies on well-defined business relationships between the QSON provider, network service providers and end users. A key challenge in making QSON a reality consists of efficiently determining routes for end user QoS flows based on the service level agreements between the QSON provider and network service providers. In this paper, we propose and present a scalable and distributed QoS routing scheme that can be used to efficiently route end user QoS flows through QSON. We demonstrate the effectiveness of our solution through simulations.

1 Introduction

With the dramatic advances in multimedia technologies and the increasing popularity of real-time applications, Quality of Service (QoS) support in the Internet has been in a great demand. However, due to many historical reasons, today's Internet primarily provides best-effort connectivity service. To enhance the current service model to provide QoS, researchers have proposed many seminal architectures represented by IntServ and DiffServ Unfortunately, realizing these QoS architectures in the Internet is unlikely to be feasible in the long run. In addition, there are no appropriate economic models for these architectures: although some ISPs might be interested in providing QoS in their own domains, there are no strong incentives for them to support QoS for users in other domains who are not their customers.

In the past few years, overlay networks have emerged as an alternative mechanism for supporting value-added services such as fault tolerance, multicasting, and security [3, 5, 12]. Many of these overlays are end-user overlays, namely, overlays are constructed purely among the end hosts without support from intermediate nodes. Due to the difficulties of supporting end-to-end QoS purely in end-user overlays, some recent work [7, 9, 15, 13, 16] proposes using backbone

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1014–1025, 2006.

overlays for QoS support, where overlays are managed by a third party provider such as an ISP.

In this paper, we adopt the approach of backbone overlays, and present a QoS overlay network architecture for scalable, efficient and practical QoS support. In this architecture, we advocate the notion of a QoS overlay network (referred to as QSON) as the backbone service domain. The design of QSON relies on well-defined business relationships between the QSON provider, network service providers (i.e., the underlying network domains which we also refer to as underlying ISPs for short), and end users: the QSON provider provisions its overlay network according to end user requests, purchases bandwidth from the network service providers based on their service level agreements (SLAs), and sells its QoS services to end users via service contracts. A key challenge in making QSON a reality consists of efficiently determining routes which satisfy the QoS requirements of end user flows based on the SLAs with the underlying ISPs. In a QSON, a QoS flow will routinely straddle multiple domains. Thus, existing QoS routing techniques which are designed primarily for flows within a single a domain may not be applicable in QSON. In this paper, we present a scalable and distributed QoS routing scheme that can be used to efficiently route end user QoS flows through QSON. We demonstrate the effectiveness of our solutions through simulations.

The layout of the paper is as follows. Section 2 describes the QSON architecture and discusses its main characteristics and advantages. Section 3 describes the routing scheme for QSON and provides a formal analysis of the scheme. Section 4 presents the simulation results to evaluate the scalability of the scheme. Section 5 provides an overview of the related work in the areas of QoS architectures and QoS routing. Section 6 offers concluding remarks and directions for future research.

2 QoS Overlay Network Architecture

In this section, we describe the QoS overlay architecture for scalable, efficient, and practical QoS support. In this architecture, a QSON (QoS Overlay Network) is constructed as the backbone service domain, which consists of many strategically deployed proxies by the QSON provider. The overlay paths between proxies are composed based on the SLAs between the QSON provider and the underlying ISPs. Outside the QSON, end hosts in access domains subscribe to the QSON by connecting to some edge proxies advertised by the QSON provider. A high level illustration of QSON is shown in Fig. 1. Though QSON is an overlay network across multiple domains, it is actually a single logical domain from the end user point of view. End user flows are managed by QSON, and the underlying ISPs only see "aggregated" flows traversing overlay paths.

2.1 Physical Network Structure

The underlying physical network structure from which QSON will be constructed is composed of multiple domains, each of which is managed by an underlying

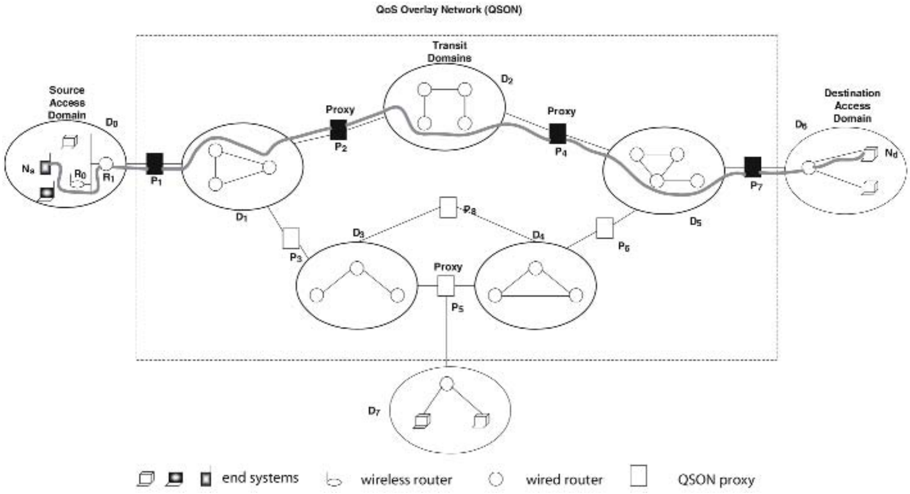


Fig. 1. The QSON Architecture

ISP. QSON proxies are strategically deployed between domains, and each proxy may belong to multiple domains. We refer to non-proxy nodes as *internal nodes*. Internal nodes can only be linked to nodes (internal nodes or QSON proxies) within its domain, whereas a QSON proxy can be linked to nodes in all the domains it belongs to. For example, in Fig. 1, QSON proxy P_2 belongs to both domain D_1 and domain D_2. We refer to domains hosting end users as *access domains*, such as domains D_0, D_6, D_7, and other domains used for data delivery as *transit domains*, such as D_1 through D_5. The QSON proxies in access domains are called *edge proxies*, which are usually advertised to end users by the QSON provider. In addition, we refer to the edge proxy in the source (or destination) access domain as *source (or destination) edge proxy*. Furthermore, the two proxies in a transit domain along a path from the source to the destination are referred to as *ingress proxy* and *egress proxy*. As shown in Fig. 1, if a connection originates in access domain D_0 and terminates in access domain D_6, P_1 is a source edge proxy, and P_7 is a destination edge proxy. Also, P_1 and P_2 are respectively the ingress and egress proxies for domain D_1.

2.2 QSON Logical Network Structure

In order to route end user QoS flows through the QSON, for each domain the QSON provider needs to know about the possible alternate paths between ingress/egress proxy pairs and the amount of bandwidth available on each one of these paths, which can be obtained from the SLAs negotiated between the QSON provider and the underlying ISP. Note that the QSON provider does not need to know the underlying topology. The logical view of the QSON thus consists of paths between pairs of proxies in each domain, and each path may be annotated by the bandwidth allocated by the ISP to the QSON. It is worth pointing out that the ISP may also provide some other information about the quality of the paths, such as the number of hops, which can be used by the QSON provider to guide the selection of one path if multiple suitable paths exist. Generally speaking, *the more information available for the possible paths in ISP domains, the better QoS support can be provided by the QSON provider to the end users.* In later sections, for the ease of presentation, we mainly use the bandwidth metric along with the number of hops to demonstrate our routing scheme. We have also investigated how the QSON can be incrementally deployed in the current "best-effort" Internet using the metrics of delay/jitter, bandwidth capacity, etc. and implemented a prototype system in PlanetLab. Due to space limit, we will not present these results in this paper. Interested readers can find more in our technical report [6].

2.3 Network State in QSON

The QSON uses the bandwidth allocated along the paths between the proxies to route end user QoS flows. As end user flows arrive and depart, the amount of available bandwidth along each one of the paths between the proxies changes dynamically. To handle such dynamic situations, each QSON proxy maintains the amount of available bandwidth on all the paths to other proxies in the same domain. Each proxy also stores a list of logical paths to the proxies in other domain(s). A logical path between the pair of proxies which do not belong to the same domain consists of a sequence of proxy nodes. To limit the amount of information to be maintained, the QSON provider can eliminate some "lengthy" paths by defining an appropriate management policy. It is usually useless to maintain very lengthy paths: the end-to-end delay may become too long and end users may not be satisfied with the service even though the available bandwidth meets the basic request. For example, in Fig. 1, proxy P_1 may know about the following logical paths $P_1P_2P_4$, $P_1P_2P_4P_7$, $P_1P_3P_5$, $P_1P_3P_5P_6$, $P_1P_3P_5P_6P_7$, $P_1P_3P_8$, $P_1P_3P_8P_6$, and $P_1P_3P_8P_6P_7$. However, the paths $P_1P_3P_5P_6P_4$ and $P_1P_3P_8P_6P_4$ may be eliminated, since these paths have twice the length (in terms of the number of logical hops) of the shortest logical path $P_1P_2P_4$. Thus, the QSON provider can pre-define a logical path length threshold lp_{th}. For a logical path between two proxies P_A and P_B, if its length is bigger than $d(1+lp_{th})$, where d is the length of the shortest logical path between P_A and P_B, then this logical path is eliminated from P_A and P_B.

2.4 Advantages of QSON

The QSON architecture combines the benefits from overlay networks and QoS-aware IP networks. On the one hand, it does not require the global deployment of QoS-aware routers. On the other hand, it can take advantage of the information obtained from intermediate nodes (proxies) to facilitate QoS support. In addition, it offers many other advantages. First, unlike end user overlays (which can only support one application), a QSON provider can support a variety of applications simultaneously. This provides an additional incentive for ISPs to adopt QSON. Second, it simplifies the management of resources in the underlying networks, since network service providers only need to provide services to a limited number of QSON providers instead of millions or billions of individual users. This is facilitated because QSON decouples the end user service management and network resource management. This level of traffic aggregation, in the long run, will make IntServ-like architectures practical.

3 Routing in QSON

In this section, we describe a distributed and scalable routing scheme, which is based on a probing technique, to efficiently route end user QoS flows in QSON.

3.1 Description of the Scheme

An end user QoS flow originates at the source node in the source access domain, traverses one or more QSON proxies in the transit domains, and terminates at the destination node in the destination access domain. In order to route such a QoS flow, an end-to-end path which satisfies the QoS constraints is obtained by composing the paths through the source and destination access domains and one or more transit domains in the QSON as explained below. In this section, we assume that the end user flow expresses its QoS constraints in terms of bandwidth for the purpose of demonstration.

Routing in Source Access Domain. To route an end user QoS flow, the source node forwards *probes* along all the existing paths to the source edge proxy within its domain. These probes are loaded with the bandwidth constraints of the flow. Each probe computes the bottleneck bandwidth of the path it traverses in the forward direction. Therefore, for each path between the source node and the source edge proxy, a probe will reach the source edge proxy. The source edge proxy then selects a suitable path that has sufficient bandwidth to satisfy the constraints of the connection. If multiple suitable paths are available, then one can be selected either randomly, or based on other criteria such as the number of physical hops along the path, or the actual bottleneck capacity of the path.

Routing Across Transit Domains. Departing from the source access domain, the probe is forwarded by the source edge proxy to other proxies in the same domain. The proxies chosen to forward the probes are such that they lie along the possible multi-domain logical paths leading to the destination edge proxy. To choose the possible multi-domain paths, we suggest a criteria of coarse-grained delay threshold based on the user request (especially if the user has explicit delay requirement): for a possible multi-domain logical path, the number of logical hops should not exceed the specified delay threshold. In this manner, the overhead incurred in forwarding the probe on paths that may not satisfy the user requirements is reduced. Before forwarding the probe to the next proxy, the source edge proxy composes the bandwidth of the path carried by the probe with the bandwidth of a suitable path between itself and the next proxy, and loads the probe with this bandwidth. A suitable path in a transit domain can be selected based on criteria similar to those described for the selection of a path in the source access domain. If no path between the chosen ingress/egress proxy pair has sufficient bandwidth to satisfy the requirement, the probe is *pruned* and not forwarded further. Otherwise it is forwarded to the next proxy along the selected path. This continues until the probe reaches the destination edge proxy.

Note that, starting from the source edge proxy, it may be likely that multiple possible logical paths leading to the destination edge proxy exist and these paths share a common portion at the beginning. For example, in Fig. 1, $P_1P_3P_5P_6P_7$, and $P_1P_3P_8P_6P_7$ share the first logical link P_1P_3 (for the case when P_1 is the source edge proxy, and P_7 is the destination edge proxy). In this case, the probe is forwarded only once along the shared path. This technique is called *probe aggregation*, which aids in the reduction of the overhead associated with forwarding the probes.

In summary, for each domain along a possible logical path, the ingress proxy forwards the probe to the egress proxy in the domain. Before forwarding the probe, the ingress proxy updates the bandwidth of the path carried by the probe with the bandwidth of a suitable path between itself and the egress proxy.

Routing in Destination Access Domain. When a probe reaches the destination edge proxy, it carries the bandwidth of a path between the source node and itself. The destination edge proxy then forwards the probe to the destination node along all the possible paths (probe flooding as in the source access domain). After receiving the probes, the destination node then selects a path based on the QoS metrics of the path(s) (i.e., the bottleneck bandwidth) and the bandwidth requirement of the connection.

3.2 An Illustrative Example

We explain the QSON routing scheme described above with the help of an example. Referring to Fig. 1, we consider a flow originating at the source node N_s in domain D_0 which is to be routed to the destination node N_d in domain D_6. The QoS constraints of the flow are expressed in terms of the required bandwidth, say b units. In order to route this flow, the source node N_s floods probes along the path $N_sR_0R_1$ towards source edge proxy P_1. At node N_s, the bandwidth of the path N_sR_0 is compared with b units, and since this bandwidth is higher than b units, the probe is forwarded to node R_0. At node R_0, the bandwidth of the path $N_sR_0R_1$ is composed, and upon determining that it is greater than b units, the probe is forwarded to node R_1. The same process is repeated at node R_1, and the probe is forwarded to the source edge proxy P_1.

At proxy P_1, three possible paths which satisfy the pre-specified delay threshold exist to the destination edge proxy P_7: $P_1P_2P_4P_7$, $P_1P_3P_5P_6P_7$, and $P_1P_3P_8$ P_6P_7. For the first path, proxy P_1 composes the bandwidth of the path $N_sR_0R_1P_1$ with the bandwidth of the path between itself and P_2, finds that the bandwidth of the entire path $N_sR_0R_1P_1P_2$ to be greater than b units and hence forwards the probe to proxy P_2. The second and the third paths share a common egress proxy P_3. As a result, a single probe is forwarded to proxy P_3 by proxy P_1 upon determining that the bandwidth of $N_sR_0R_1P_1P_3$ is greater than b units.

For the first path $P_1P_2P_4P_7$, when the probe reaches proxy P_2, P_2 determines that the path between N_s and P_4 satisfies the bandwidth constraints and forwards the probe to proxy P_4. For the second and third paths, however, the probe is pruned because the bottleneck bandwidth of the paths between P_3 and the egress proxies P_5 and P_8 is not sufficient.

The probe that reaches proxy P_4 continues to traverse to the destination proxy P_7. When a probe reaches the destination proxy P_7, probes are once again flooded to the destination node N_d along all the existing paths.

3.3 Correctness and Complexity Analysis

Complexity Analysis. Given an overlay network $G = (V, E)$, where V is the set of QSON proxies, and E is the set of logical links connecting QSON proxies. Let $|V| = n$ and $|E| = m$. For each logical link $e_i \in E$ $(1 \leq i \leq m)$, it is assigned a value ph_i, which represents the number of physical paths connecting the two QSON proxies of e_i. We assume ph_i is bounded by ph_B. The diameter of G, i.e., the length of the longest shortest logical path between any pair of QSON proxies, is represented by D. Further, we denote the pre-defined logical path length threshold (compared with the shortest logical paths) as lp_{th} and the maximum number of logical paths between any pair of proxies as W.

In the proposed QSON routing scheme, for any end user QoS flow, the number of probe messages can be bounded by $WD(1 + lp_{th})$, where $D(1 + lp_{th})$ denotes the maximum length of a logical path. To set up a reference point, we choose an intuitive inter-domain QoS distributed routing scheme, in which probes are flooded along all paths across the domains. We refer this approach to as *all-path-flood inter-domain distributed routing* or *all-path-flood routing* in short. In this algorithm, the number of probe messages has an upper bound of $Wph_B^{D(1+lp_{th})}$[1]. This simple analysis demonstrates the dramatic difference between these two schemes: the control message overhead of the QSON routing scheme increases much more slowly to the delay threshold lp_{th} than that of the all-path-flood routing scheme. Note that the probe messages included here are only those inside QSON, i.e., in transit domains, and they do not include probe messages in access domains, which are actually the same for both QSON routing and all-path-flood routing. In fact, in QSON routing, the probe overhead can be reduced further by employing probe aggregation and pruning techniques. In Section 4, we will evaluate the scalability of QSON routing using simulations.

Correctness Analysis. *Claim 1: Given the logical path length threshold lp_{th}, if there exits one path lp_1 which satisfies the end user request (say, b units of bandwidth), then QSON routing will find at least one suitable path.*

This claim can be easily proved as follows: If we assume QSON routing can not find any path for the end user QoS flow, then the probe along the path lp_1 must have been pruned (probe pruning is the only possible exit point for a probe before reaching the destination). However, according to the given property: lp_1

[1] Various techniques have been developed to improve the performance of the all-path-flood routing algorithm. However, unless a similar hierarchical routing structure to that of QSON is adopted, its routing overhead can not be reduced significantly by orders of magnitude.

satisfies the end user request, i.e., the bottleneck bandwidth along lp_1 is greater than b. In order words, the probe cannot be pruned along the path lp_1, as it conflicts with the above premise. Thus, QSON routing will find at least one suitable path.

4 Performance Evaluation

In this section, we evaluate the scalability of QSON routing via simulations. We compare QSON routing with all-path-flood inter-domain distributed routing. Further, we investigate how the probe aggregation and pruning techniques help to improve the performance.

4.1 Simulation Settings

We implement the QSON routing in a simulator built using C++. In the simulations, we use two types of network topologies, a real AT&T backbone network with 120 nodes [1] and 10 random networks with 100 nodes generated using the Waxman model [17]. Each node in the topologies represents a proxy, and each edge denotes a logical link between two proxies in the same domain. We use the delay threshold lp_{th} to compute the logical paths: a logical path from a source proxy to a destination proxy should be no more than lp_{th} logical hops longer than the shortest logical path between them. We vary the delay threshold from 0 to 4 logical hops.

We assume that the number of physical paths connecting two neighbor proxies follows a uniform distribution in the range of $[1, 10]$. To evaluate the effectiveness of the probe pruning technique, we set the available bandwidth between two neighbor proxies to be uniformly distributed between 1 and 20 units. Unless otherwise specified, an end user QoS flow requires 10 units of bandwidth. For each topology, we generate 1000 QoS flows in each simulation run. For each flow, the source and the destination are chosen randomly from all the nodes in the network. We conduct 1000 simulation runs by varying the available bandwidth of the logical paths. The results are averaged over the 1000 runs.

We use *control overhead* as the performance metric to evaluate the scalability of QSON routing. It is defined as the total number of logical hops that probes traverse. The lower the control overhead, the less the consumed bandwidth, and hence the more efficient the scheme.

4.2 Results and Analysis

We conduct three sets of simulation experiments to examine the performance of QSON routing, the probe aggregation and pruning techniques, respectively.

QSON Routing. We plot the results of QSON routing without probe aggregation and pruning vs. all-path-flood routing (denoted as "Non-QSON") for the AT&T backbone and Waxman topologies in Fig. 2 and 3, respectively (Note that the control overhead is plotted in the log scale). These two figures show

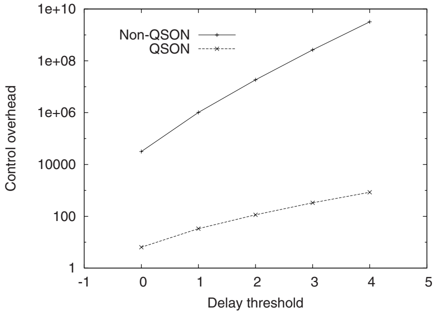


Fig. 2. QSON vs. Non-QSON routing in the AT&T topology

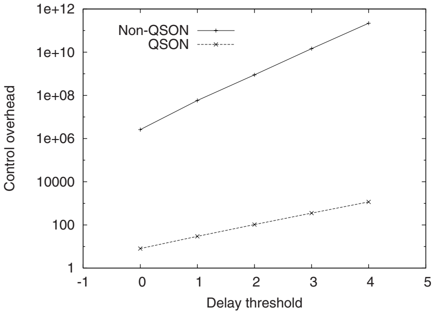


Fig. 3. QSON vs. Non-QSON routing in the Waxman topologies

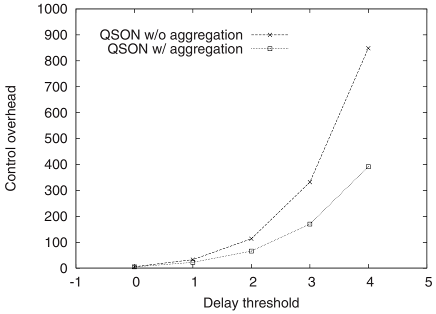


Fig. 4. Effectiveness of the probe aggregation technique in the AT&T topology

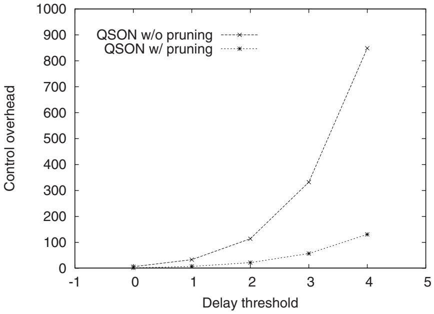


Fig. 5. Effectiveness of the probe pruning technique in the AT&T topology

the same trends. First, it is clear that QSON routing effectively reduces the control overhead by sending only one probe between every involved ingress/egress pair. For instance, when the delay threshold is 4, QSON decreases the control overhead from 3.20×10^9 to 848 in the AT&T topology, and from 2.2×10^{11} to 1166 in the Waxman topologies, both corresponding to more than 99% overhead reduction.

Second, as the delay threshold increases, more logical paths between neighbor proxies become eligible. Consequently, both routing schemes result in a higher control overhead. However, comparing the increasing trend of the two curves in each figure, we observe that the overhead of all-path-flood routing grows almost exponentially with the delay threshold, whereas that of QSON routing grows much less rapidly. This difference is caused by the uncontrolled flooding involved in the former scheme, and it is indeed consistent with our analysis in Section 3.3.

Probe Aggregation. The results of QSON with and without the probe aggregation technique in the AT&T topology are shown in Fig. 4. This figure demonstrates that probe aggregation helps reduce the control overhead of QSON routing. In addition, the reduction in the overhead increases with the delay threshold, since more logical paths are likely to share a larger common portion at the beginning of these paths. For example, when the delay threshold is 0, probe aggregation reduces the control overhead by less than 10%; when the threshold is raised to 4, it decreases more than half of the overhead. The results for Waxman graphs are similar and thus are not shown here to save space.

Probe Pruning. Fig. 5 illustrates the benefit of using the probe pruning technique in the AT&T backbone when the QoS flows require 10 units of bandwidth. Obviously, this technique improves the performance of QSON routing with respect to control overhead: when the delay threshold is varied, probe pruning consistently decreases the overhead by more than 70%. Furthermore, this benefit becomes very distinguished when the delay threshold is higher: approximately 85% of the control overhead is reduced at a delay threshold of 4. In this case, the logical paths tend to go through a larger number of proxies and it is more likely that some portions of these paths can be pruned due to bandwidth deficiency.

Summary. The results of our simulations reported in this section indicate that QSON routing generates significantly less control overhead than all-path-flood routing. In addition, both the probe aggregation and pruning techniques are very effective in further reducing the control overhead, especially when the delay threshold is high.

5 Related Work

In this section, we briefly review some related QoS overlay architectures and QoS routing schemes along with their pros and cons.

5.1 QoS Overlay Architectures

Recently, overlay networks are proposed to support value-added services without making changes to network routers. End-user overlays rely on end systems to implement QoS features. For example, Resilient Overlay Networks (RONs) are proposed to detect and recover from Internet path failures by actively monitoring the quality of overlay links and routing packets based on application-specific metrics [3]. The Spine architecture applies TCP-like loss recovery and congestion control on each overlay link to reduce the latency and jitter of reliable connections [2]. Though highly flexible, end-user overlays usually cannot provide end-to-end QoS guarantees, since they normally cross many uncontrolled intermediate domains. Moreover, it is difficult to design an effective economic model for ISPs to adopt end-user overlays.

To solve these problems, backbone overlays managed by third party providers are advocated. Some example proposals are Service Overlay Network or SON [7], OverQoS [15], QRON [13], and QUEST [9]. Unlike QSON, which well combines the benefits of overlay networks and QoS-aware IP networks, these proposals either assume the guaranteed services from underlying networks, such as SON [7], QRON [13], and QUEST [9], or try to infer the statistical bandwidth and loss rate assurance along overlay links, e.g., OverQoS [15]. Moreover, none of these proposals addresses the QoS routing issue in backbone overlays, which in fact is the main problem of this paper.

5.2 QoS Routing

QoS routing techniques can be categorized into source routing and distributed routing. In the former case, the source node is responsible for determining a suitable path by applying graph algorithms to the link state information stored at the source node [4]. A link state protocol is used to broadcast link state information through the network, which consumes an enormous amount of resource. On the other hand, distributed routing is achieved by probe flooding, where the source node floods probes towards the destination node searching for suitable paths [4]. If multiple suitable paths satisfying the constraints exist, then the shortest path is chosen to reduce delay and the probability of path degradation.

When used for inter-domain routing in large networks, the overhead associated with source and distributed routing is even aggravated. Aggregation techniques for source routing have been proposed to alleviate this issue [11], but it may lead to inaccuracies, crankback, and reaggregation [10, 8]. The scalability of inter-domain routing via probe flooding can be improved by precomputing only the shortest paths [14]. However, even the number of shortest paths in the case of a large network is likely to be very high. When a QoS flow is to be routed through QSON, it will typically cross multiple domains. If aggregation techniques are used, they will lead to inaccurate and sub-optimal solutions especially when the resource availability is low. If distributed routing via probe flooding is to be used, then flooding probes across all the possible paths will consume resources that could be otherwise used for supporting additional flows. Due to these reasons, the existing QoS routing techniques cannot be used for routing end user flows through QSON.

6 Conclusions and Future Work

In this paper, we presented a backbone QoS overlay network (QSON) architecture for scalable, efficient and practical QoS support. The major contributions of this paper can be summarized as follows. (1) We advocate backbone overlays for scalable QoS support, and present an integrated QSON architecture which involves access domains and transit domains. (2) We propose a scalable and distributed QSON routing scheme, which can reduce the probe overhead significantly compared with all-path-flood routing. (3) We conduct simulations to evaluate the performance of QSON routing, and the results show that its probe overhead is significantly reduced by more than 99% in the simulated scenarios.

We plan our future work in the following two directions. (1) Network design for QSON: In this paper, we assume the overlay network is known. We do not consider the overlay network design, i.e., where to place proxies and which logical links to select. Clearly, overlay network design for QSON is a challenging issue to investigate. (2) Comparison with various multihoming proposals: Multihoming route control is closely related to overlay routing. It is worth investigating how QSON performs compared with multihoming routing schemes.

References

1. AT&T IP Backbone. *http://www.ipservices.att.com/backbone/*, 2001.
2. Y. Amir and C. Danilov. Reliable communication in overlay networks. In *Proceedings of the IEEE International Conference on Dependable Systems and Networks, June 2003.*
3. D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceedings of Symposium on Operating Systems Principles*, pages 131–145, 2001.
4. S. Chen and K. Nahrstedt. An overview of Quality–of–Service routing for the next generation high–speed networks: Problems and solutions. *IEEE Network, Special Issue on Transmission and Distribution of Digital Video*, 1998.
5. Y.-H. Chu, S. G. Rao, and H. Zhang. A case for end system multicast. In *Proceedings of ACM Sigmetrics*, June 2000.
6. J.-H. Cui, S. Gokhale, L. Lao, and J. Lu. Distributed QoS Routing for Backbone Overlay Networks. UCONN CSE Technical Report: UbiNet-TR06-01, Jan. 2006.
7. Z. Duan, Z.-L. Zhang, and Y. T. Hou. Service overlay networks: SLAs, QoS, and bandwidth provisioning. *IEEE/ACM Transactions on Networking*, 11(6):870–883, 2003.
8. E. Felstaine, R. Cohen, and O. Hadar. Crankback prediction in hierarchical ATM networks. In *Proc. of INFOCOM*, 1999.
9. X. Gu, K. Nahrstedt, R. Chang, and C. Ward. Qos-assured service composition in managed service overlay networks. In *Proceedings of IEEE 23rd International Conference on Distributed Computing Systems, Providence, May 2003.*
10. R. Guerin and A. Orda. QoS–based routing in networks with inaccurate information: Theory and algorithms. In *Proc. of INFOCOM*, 1997.
11. F. Hao and E. Zegura. On scalable QoS routing: Performance evaluation of topology aggregation. In *Proc. of INFOCOM*, 2000.
12. A. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure Overlay Services. In *Proceedings of ACM SIGCOMM'02, (Pittsburgh, PA), August 2002.*
13. Z. Li and P. Mohapatra. QRON: QoS-aware routing in overlay networks. *IEEE Journal on Selected Areas in Communications, January*, 2004.
14. S. Norden and J. Turner. Inter–domain QoS routing algorithms. Technical Report WUCS-02-03, Department of Computer Science, WUCS, 2002.
15. L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. Overqos: An overlay based architecture for enhancing internet qos. *In Proceedings of USENIX NSDI'04*, pages 71–84, 2004.
16. S. Vieira and J. Liebeherr. Topology design for service overlay networks with bandwidth guarantees. In *IEEE IWQoS*, 2004.
17. B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988.

Distributed Linear Time Construction of Colored Trees for Disjoint Multipath Routing[*]

Srinivasan Ramasubramanian, Mithun Harkara, and Marwan Krunz

Department of Electrical and Computer Engineering,
University of Arizona, Tucson, AZ 85721

Abstract. Disjoint multipath routing (DMPR) is an effective strategy to achieve robustness in networks, where data is forwarded along multiple link- or node-disjoint paths. DMPR poses significant challenges in terms of obtaining loop-free multiple (disjoint) paths and effectively forwarding the data over the multiple paths, the latter being particularly significant in datagram networks. One approach to reduce the number of routing table entries for multipath forwarding is to construct two trees, namely red and blue, rooted at a destination node such that the paths from a source to the destination on the two trees are link/node-disjoint. This paper develops the first distributed algorithm for constructing the colored trees whose running time is linear in the number of links in the network. The paper also demonstrates the effectiveness of employing generalized low-point concept rather than traditional low-point concept in the DFS-tree to reduce the average path lengths on the colored trees.

1 Introduction

Multipath routing (MPR) is an effective strategy to achieve robustness [1], load balancing [2], congestion reduction [3], low power consumption [4], and increased throughput. It operates by transmitting data over multiple paths. In general, the multiple paths from a source to a destination may have common links (or nodes) as long as the shared links (or nodes) have sufficient resources. To improve the transmission reliability and avoid shared-link (or node) failures, the multiple paths can be selected to be link- or node-disjoint. In this case, the MPR approach is referred to as *disjoint multipath routing* (DMPR). DMPR provides better robustness compared to the generic MPR. However, it may be inefficient with respect to other metrics such as the overall energy consumption [4] in a wireless ad hoc or sensor network.

DMPR has been extensively studied in the context of wired networks [5, 6], where the multiple paths are often employed for failure resiliency purposes. Only one of the paths, referred to as the primary path, is used at any instant. Upon a failure, the connection is rerouted over a backup path. If the backup path is the same for any link (or node) failure that affects the primary path, then the primary and backup paths must be link- (or node-) disjoint. In applications such as transmission of multiple description encoded video streaming, the two link-disjoint paths are used simultaneously. Two independently encoded video streams are transmitted along two link-disjoint paths [7]. If

[*] The research developed in this paper is supported by National Science Foundation under grants 0325979, 0435490, and EEC-0333046.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1026–1038, 2006.

multiple paths are employed for increased throughput, then the data may be split over multiple paths.

Motivation. Implementation of generic MPR and DMPR poses two main challenges. The first is related to the computation of loop-free multiple paths. Several centralized algorithms (or equivalently those that assume a global network knowledge) have been proposed for the DMPR problem in the context of failure resiliency in wired connection-oriented networks. For large-scale wired or wireless networks, a distributed solution that relies only on local information is preferred. Distributed multipath routing algorithms in the literature are developed purely in the context of wireless networks. MPR approaches based on Dynamic Source Routing (DSR) [8, 9, 10] require the destination to select maximally disjoint paths among the received route requests. MPR approaches based on AODV routing [11, 12, 13, 14, 15] do not guarantee finding disjoint paths. The only well-known generic multipath routing employed in the wired datagram network is the OSPF algorithm, where the choice of paths is limited to those of equal cost.

The second challenge of implementing MPR (or DMPR) techniques is related to forwarding of data over the multiple paths. In typical connection-oriented networks, the end-to-end path is clearly identified using, for example, connection identifiers or labels. The nodes maintain a routing table that specifies the output port for each label. Each path requires a unique identifier. Hence, the size of the routing table at each node is directly proportional to the number of multiple paths. In contrast, datagram networks rely on the destination address in the packet header for forwarding packets over one path. To implement MPR or DMPR techniques in such networks, every node must maintain a set of preferred neighbors to reach a destination, such that the paths are loop-free (and disjoint, if needed). Forwarding of packets to meet such constraints must be based on destination address and some “additional” information (e.g. source address, labels, etc.). The intermediate nodes must be aware of this additional information or otherwise, it must be carried in every packet header. The choice of the additional information used in forwarding along multiple paths determines the overhead involved.

To reduce the routing table overhead, hence reduce lookup time, a novel multipath routing strategy called *colored trees* (CT) was developed [16]. Every node in the network has two preferred neighbors to the destination: *red* and *blue*. A packet transmitted from a source is marked with one of the two colors. An intermediate node that receives the packet forwards it to its preferred neighbor based on the color of the packet. Thus, the routing table at a node has only two entries (for every destination node). The network may be viewed as two trees (red and blue) that are rooted at the drain. The two paths from a given source to the drain on the two trees are link/node-disjoint.

The goal of this paper is to develop a linear-time distributed algorithm for constructing two colored trees. The rest of the paper is organized as follows. Section 2 describes the network model and problem definition. Section 3 discusses the related work in obtaining colored trees using generalized path augmentation technique and maintaining the partial order in a distributed manner using local information. Section 4 develops the linear-time distributed algorithm for constructing the two colored trees. Section 5 presents the performance comparison of the distributed algorithm with traditional and generalized low-point concepts. Our Conclusions are presented in Section 6.

2 Problem Statement

Consider a network $\mathcal{G}(\mathcal{N}, \mathcal{L})$ composed of a set of nodes $\mathcal{N}$ and a set of links $\mathcal{L}$. The links are assumed to be bi-directional. The terminology of *arc* is used to refer to a directed link between two nodes. An arc from node i to j is represented as $i \rightarrow j$. Given a drain node $d \in \mathcal{N}$, the goal is to construct two trees $\mathcal{R}$ and $\mathcal{B}$ (referred to as the red and blue trees, respectively) rooted at d that minimize the average path length from a source to the drain such that the CT-LD and CT-ND versions of the problem satisfy the link-disjoint and node-disjoint path constraints, respectively. These constraints are stated as follows. Let $\mathcal{P}_{sd}^{\mathcal{R}}$ and $\mathcal{P}_{sd}^{\mathcal{B}}$ denote the paths from a node s to the drain d on trees $\mathcal{R}$ and $\mathcal{B}$, respectively.

Link-disjoint path constraint: $\forall s \in \mathcal{N} \setminus \{d\}$ and $\forall i, j \in \mathcal{N}$

$$i \rightarrow j \in \mathcal{P}_{sd}^{\mathcal{R}} \;\;\Rightarrow\;\; (i \rightarrow j \notin \mathcal{P}_{sd}^{\mathcal{B}}) \wedge (j \rightarrow i \notin \mathcal{P}_{sd}^{\mathcal{B}}).$$

Node-disjoint path constraint: $\forall s \in \mathcal{N} \setminus \{d\}$ and $\forall i \in \mathcal{N} \setminus \{s, d\}$

$$i \in \mathcal{P}_{sd}^{\mathcal{R}} \;\;\Rightarrow\;\; (i \notin \mathcal{P}_{sd}^{\mathcal{B}}).$$

A network must be two-node-connected (two-edge-connected) to obtain a solution to the CT-ND (CT-LD) problem [17].

3 Generalized Path Augmentation

The generalized path augmentation algorithm [18] is a heuristic developed in the context of robust multicasting. It may be applied to the problem at hand by simply reversing the direction of the links in the trees constructed. It starts by choosing an arbitrary directed cycle $(d, v_1, ..., v_k, d)$ in $\mathcal{G}$ with at least three nodes ($k \geq 2$). If this cycle does not include all the nodes of $\mathcal{G}$, then a path that starts and ends on that cycle and that passes through at least one node not on the cycle is chosen for augmentation. The algorithm continues with path augmentation until all the nodes in the network are considered.

Medard et al. [17] developed a centralized algorithm that selects a cycle and successive paths at random. Xue et al. [18] developed a generalized version of the centralized path augmentation approach (referred to as the XCT algorithm in the rest of the paper) by specifying certain criteria for selecting paths for augmentation, which depend on the problem objective (e.g. minimizing average delay or cost, maximizing bandwidth, etc.).

The XCT algorithm is based on partial ordering of nodes in the network. The partial order $\prec$ of the nodes on the blue tree $\mathcal{B}$ is defined as follows. If $u \rightarrow v \in \mathcal{B}$, then v precedes u in the partial order, represented as $v \prec u$ (the algorithm in [18] employs partial order on both the red and blue trees for link-disjoint paths. However, the explanation here has been simplified based on the partial order on the blue trees and node-disjoint paths). The partial ordering satisfies the transitive relationship, i.e., if $u \prec v \prec w$, then $u \prec w$. The generalized approach is now described for the construction of two colored trees for the CT-ND problem.

The XCT algorithm for constructing two trees that satisfy the node-disjoint path constraint follows four steps:

1. Initialize $\mathcal{R}$ and $\mathcal{B}$ to contain the root node d only. Initialize the partial order of the nodes to be the empty set.
2. Find a cycle $(d, v_1, ..., v_k, d)$. Let $v_k \rightarrow v_{k-1} \rightarrow ... \rightarrow v_1 \rightarrow d$ be the *red chain* and $v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_k \rightarrow d$ be the *blue chain*. Add the blue chain to $\mathcal{B}$ and the red chain to $\mathcal{R}$. Update the precedence relation with $v_1 \prec v_2 \prec ... \prec v_k \prec d$.
3. Stop if $\mathcal{B}$ spans all the nodes in $\mathcal{G}$.
4. Find a path $(x, v_1, ..., v_k, y)$ that connects any two distinct nodes x and y on $\mathcal{B}$ and any k nodes not on $\mathcal{B}$, $k \geq 1$, such that $x \prec y$. Let $v_k \rightarrow v_{k-1} \rightarrow ... \rightarrow v_1 \rightarrow x$ be the red chain and $v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_k \rightarrow y$ be the blue chain. Add the blue chain to $\mathcal{B}$ and the red chain to $\mathcal{R}$. Update the precedence relation with $x \prec v_1 \prec v_2 \prec ... \prec v_k \prec y$. Go to Step 3.

The above algorithm may be applied to the link-disjoint case by relaxing the condition in Step 4 that x and y have to be distinct and maintaining partial ordering of edges instead of nodes. The algorithm is guaranteed to obtain two trees that satisfy the link-disjoint (node-disjoint) constraint if the network is two-edge-connected (two-node-connected). The approach may be combined with depth-first-search numbering to obtain an $O(L)$ algorithm to construct the colored trees [19].

The algorithms developed in [18] and [19] assume a complete knowledge of network topology; i.e., they are centralized algorithms. For large networks, a distributed implementation is essential. In such a distributed implementation, nodes are assumed to have only neighborhood information.

3.1 Maintaining the Partial Order in a Distributed Fashion

The crux in developing such a distributed algorithm is to identify a mechanism to manage the partial order in a distributed fashion, where each node relies only on local information. Consider the example network in Figure 1.

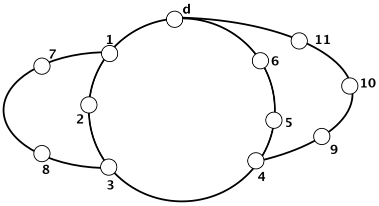


Fig. 1. Example network to illustrate partial ordering and path augmentation used to develop the distributed colored-tree construction algorithm

Let the first cycle selected by the centralized algorithm be $(d, 1, 2, 3, 4, 5, 6, d)$. Considering one particular direction in the cycle (corresponding to say the blue tree), the partial ordering of the nodes would be $1 \prec 2 \prec 3 \prec 4 \prec 5 \prec 6 \prec d$. There are two options for selecting a path for augmentation: 1–7–8–3 or 4–9–10–11–d. The algorithm by Medard et al. selects a path at random while that by Xue et al. selects a path based on a certain metric. Without loss of generality, assume that the path 4–9–10–11–d is chosen for augmentation. The partial ordering of these paths must be such that: (1) node 4 precedes node 9 ($4 \prec 9$); node 11 precedes node d ($11 \prec d$); and nodes 9, 10, and 11 must appear in the same order as in the path ($9 \prec 10 \prec 11$). However, it is to be noted that there is no explicit ordering between the nodes 9, 10, and 11 in the new path and the nodes 5 and 6 in the old path. Some of the valid global ordering of the nodes that satisfy the above partial order are:

1. $1 \prec 2 \prec 3 \prec \underline{4} \prec 5 \prec 6 \prec \mathbf{9} \prec \mathbf{10} \prec \mathbf{11} \prec \underline{d}$
2. $1 \prec 2 \prec 3 \prec \underline{4} \prec \mathbf{9} \prec \mathbf{10} \prec \mathbf{11} \prec 5 \prec 6 \prec \underline{d}$
3. $1 \prec 2 \prec 3 \prec \underline{4} \prec \mathbf{9} \prec 5 \prec \mathbf{10} \prec 6 \prec \mathbf{11} \prec \underline{d}$

It is the choice of which of these global orderings is selected that distinguishes various approaches. In order to develop a distributed algorithm employing only local information, we select the first ordering, namely $1 \prec 2 \prec 3 \prec \underline{4} \prec 5 \prec 6 \prec \mathbf{9} \prec \mathbf{10} \prec \mathbf{11} \prec \underline{d}$. Given that the first cycle is formed, the global ordering of the nodes is fixed. The path selection starts from the node that is the highest in the order. If a new path can be selected for augmentation from the highest node, then such a path is chosen. The nodes in the new path are added to the global order just before the node from which the path was computed. Once the highest node exhausts all possibilities (adding paths through each of its neighbor), then the path search begins with the next node in the list.

4 Linear-Time Distributed Construction of Colored Trees

The linear-time distributed algorithm for constructing the colored trees works in two phases: (1) Distributed DFS numbering and generalized low-point computation, and (2) Distributed path augmentation. The distributed algorithm is sequential in nature and requires only neighborhood information.

4.1 Distributed DFS Numbering and Generalized Low-Point Computation

We assign DFS numbers to the nodes in the network starting from the drain. The drain is assigned the DFS number 1. In order to help compute paths for augmentation without backtracking, we compute the *generalized low-point value* of a node.

The *low-point value* of a node n is traditionally defined as the lowest DFS-index of a node that can be reached from n by using DFS-tree[1] edges and at most one back edge. The *low-point path* of node n is the path traversed to reach the low-point node. The low-point path of a node n is of the form $n \rightarrow i_1 \rightarrow i_2 \rightarrow \ldots \rightarrow i_k \rightarrow n'$ ($k \geq 0$) such that: (1) node n is the DFS-parent of node i_1, (2) node i_{j-1} is the DFS-parent of node i_j ($2 \leq j \leq k$), (3) the DFS-index of n' is lower than that of n; and (4) the DFS-index of n' is the lowest among all such possible paths. The algorithm developed in [19] employs the traditional low-point value and path.

[1] A DFS-tree is a tree rooted at the drain and the arcs in the tree are directed away from the drain. A back edge is an edge that connects a higher DFS-index node to a lower DFS-index node. The *low-point node* of a node n is the node whose DFS-number is the LPV of node n.

We define the *generalized low-point value* (GLPV) of a node n as the lowest DFS-index of a node that can be reached from node n by traversing a sequence of nodes with increasing DFS-index with the exception of the last hop. The *generalized low-point path* of a node n is of the form $n \rightarrow i_1 \rightarrow i_2 \rightarrow \ldots \rightarrow i_k \rightarrow n'$ $(k \geq 0)$, such that: (1) the DFS-index of n is lower than that of i_1, (2) the DFS-index of i_{j-1} is lower than that of i_j $(2 \leq j \leq k)$, (3) the DFS-index of node n' is lower than that of node n, and (4) the DFS-index of n' is the lowest among all such possible paths. The *generalized low-point neighbor* (GLPN) of a node n is defined as that neighbor of node n which is on its generalized low-point path.

The GLPV and GLPN of a node are computed during the distributed DFS numbering phase. The algorithm to assign the DFS-indices and compute the GLPV and GLPN is shown in Figure 2. The DFS-indices of all the nodes are initialized to -1. We incorporate hop count as a metric to compute the shortest generalized low-point path among those available. Note that the linear-time algorithm developed in this paper will work with the traditional low-point of a node, however, the path length optimization cannot be made as the arcs are forced to be on the DFS-tree, except the last hop.

Notation	Comment
dfs[n]	DFS-index of node n.
dfsparent[n]	DFS-parent of node n.
glpv[n]	Generalized low-point value of node n.
glpn[n]	Generalized low-point neighbor of node n.
glpd[n]	Generalized low-point distance (hop count) of node n.

```
DFS(parent, n, currdfs)
1.      if $dfs[n] > 0$ return currdfs;
2.      $dfs[n] = currdfs$; $dfsparent[n] = parent$; $currdfs = currdfs + 1$;
3.      for every neighbor $i \neq parent$ of n do:
3.A.        $currdfs$ = DFS(n, i, currdfs);
3.B.        if $(dfs[i] < dfs[n])$ and $(dfs[i] \leq glpv[n])$
3.B.i.          $glpv[n] = dfs[i]$; $glpn[n] = i$; $glpd[n] = 1$;
3.C.        else if $(dfs[i] > dfs[n])$ and $(glpv[i] < glpv[n])$
3.C.i.          $glpv[n] = glpv[i]$; $glpn[n] = i$; $glpd[n] = glpd[i] + 1$;
3.D.        else if $(dfs[i] > dfs[n])$ and $(glpv[i] = glpv[n])$ and $(glpd[i] < glpd[n] - 1)$
3.D.i.          $glpn[n] = i$; $glpd[n] = glpd[i] + 1$;
4.      return currdfs;
```

Fig. 2. Algorithm to assign DFS-indices to the nodes and compute generalized low-point value and neighbor of a node

Given a two-edge-connected network, the GLPV of a node n is lower than or equal to the DFS-index of its DFS-parent. Given a two-node-connected network, the GLPV of a node n is strictly lower than the DFS-index of its DFS-parent. The GLPV provides a mechanism to identify if the network is two-edge or two-node connected for reaching the drain in linear time. Every node sends a DFS message to all its neighbors (except its parent) and receives a DFSRETURN message in response. The number of DFS and

DFSRETURN messages sent are $2|\mathcal{L}| - (|\mathcal{N}| - 1)$ each. At the end of the distributed DFS-numbering phase, every node in the network is aware of the DFS numbers of its neighbors. The neighbors of a node are arranged in an increasing order of their DFS-indices.

4.2 Distributed Path Augmentation

An overview of the steps involved in the distributed path augmentation is shown in Figure 3. The drain is initialized to the TOKEN state, indicating that it is already added to the trees and has the authority to initiate path search. The other nodes are initialized to the UNVISITED state.

Distributed Path Augmentation Algorithm

1. Arrange the neighbors in the neighbor list in an increasing order of their DFS-indices.
2. On receiving a TOKEN message, initiate path search along every node in the neighbor list, one at a time.
   (a) Every node that receives the SEARCH message forwards it sequentially to every node in the neighbor list according to some forwarding rules.
   (b) When a SUCCESS message returns from a neighbor, the value of msg.flagNewNodeAdded is stored in the neighbor list.
3. Forward the TOKEN message to every node if the flagNewNodeAdded flag for this neighbor is TRUE. The neighbor list is traversed in the reverse direction. Every node finishes its operation and sends a RETURN message back.
4. After receiving a RETURN message from all the neighbors to whom the token message was sent, send RETURN message to the parent that sent the TOKEN message.

Fig. 3. Overview of the steps involved in the distributed algorithm for computing colored trees

Path search. The drain initiates a path search sequentially along its neighbors. The first search is for a cycle while the others are for a path. On receiving a SEARCH message, a node in the UNVISITED state changes itself to the VISITED state, which indicates that the node is part of the path being chosen for augmentation. The SEARCH message is then forwarded to one of the neighbors (based on the forwarding rules discussed later). The drain always responds to a SEARCH message with a SUCCESS. When a SEARCH message reaches any other node, that node responds with a SUCCESS message if it is in the CYCLE state[2]. If a node in the TOKEN state, it is configured to send a SUCCESS message, then the paths for augmentation may start and end at the same node, resulting in a solution for the CT-LD case. If the node in the TOKEN state is configured to respond with a FAILURE, then the result would be a solution for the CT-ND case.

A node in the CYCLE state responds to a SEARCH message with a SUCCESS message in which msg.flagNewNodeAdded is set to FALSE, indicating no new nodes are added further down the path. This enables the receiving node to not forward the search token to

[2] A node in the CYCLE state indicates that it has already been added to the colored trees. It has not received the TOKEN message to initiate path search.

a node that is already on the cycle. As a rule, *a path search token may be forwarded from node i to node j only if node j was added to the path through a path search message from node i to j.* Note that as paths are being searched from the highest node in the global-order list (maintained in a distributed manner), any node that is on the cycle that receives the search message must be lower in the global-order list than the node that initiated the message. Upon receiving a SUCCESS message, a node in the VISITED state changes to the CYCLE state. It adds the node from which it received the SEARCH message as its parent on the blue tree and the node from which it received the SUCCESS as its parent on the red tree. The flagNewNodeAddedvariable for that neighbor is set to the value indicated in the message. The node then sends a SUCCESS message to the node from which it received the SEARCH message with the msg.flagNewNodeAdded set to true, indicating that it was newly added to the path.

Forwarding search token. A node that has the path search token attempts to augment a path through each of its neighbor. The node then forwards the token to those eligible neighbors, traversing the ordered list in the reverse direction (opposite to the order in which the SEARCH messages were initiated), one at a time. An eligible neighbor is one for which the variable flagNewNodeAdded is set to true. Such an order reversal for passing the token helps maintain a consistent global ordering in a distributed manner across all the nodes in the network. A node that receives a TOKEN changes its state from CYCLE to TOKEN, starts the path search along each of its neighbors, and forwards the token to its eligible neighbors.

Once the tokens are returned by all neighbors, the node sets its state to FINISH and returns the token to the node from which it first received the token. The token finally reaches the drain, indicating that all nodes in the network are in the FINISH state, at which point the algorithm terminates.

4.3 Forwarding Rules for Path Augmentation Without Backtracking

The cycle and paths required for the distributed path augmentation approach are computed using four types of messages. A node sends out a SEARCH message to obtain a path (or cycle) for augmentation. In order to obtain a path in a distributed fashion without backtracking, we develop certain forwarding rules with some additional information in every message.

Let a node x, which has been already added to the trees, attempt a path search by sending a message through its neighbor y. Every search message msg contains the following fields: (1) msg.source is the source of the message, (2) msg.sourcedfs is the DFS-index of the source; (3) msg.specialFlag is a flag based on which the message may be routed to a different neighbor other than the default lowest DFS-index neighbor; and (4) msg.glpv is the GLPV of the source that initiated the message which could be modified at an intermediate node. A SEARCH message initiated by node x has the msg.specialFlag set to DEFAULT. If node y is not added to the colored trees, it forwards the message to one of its neighbors according to the rules shown in Figure 4.

Case 1: $\texttt{dfs}[\texttt{x}] > \texttt{dfs}[\texttt{y}]$.
In this case, there exists a path from y to the drain by successively traversing the lowest DFS-index neighbor from y to reach the drain. As the drain is already a part of the

Notation	**Comment**
msg	Message received by node y.
msg.source	Source node of message msg.
msg.sourcedfs	DFS-index of the source node of message msg.
msg.specialFlag	Special flag field in the message.
msg.glpv	Generalized low-point value indicated by a node in the message.
newmsg	Message sent by node y.

Rules to forward a message.

```
1.        newmsg.source = y;  newmsg.sourcedfs = dfs[y];
2.            if (msg.specialFlag = PARENTFLAG)
2.A.              z = lowpoint[y];
2.B.              if (dfs[z] = glpn[y]) newmsg.specialFlag = DEFAULT;
2.C.              else newmsg.specialFlag = PARENTFLAG;
3.            else if (msg.specialFlag = LOWPOINTFLAG)
3.A.              if (glpv[y] < msg.glpv)
3.A.i.                z = glpn[y]; newmsg.specialFlag = PARENTFLAG;
3.B.              else
3.B.i.                z = dfsparent[y];
3.B.ii.               newmsg.specialFlag = LOWPOINTFLAG; newmsg.glpv = msg.glpv;
4.            else
4.A.              if the lowest DFS-index neighbor is not the same as msg.source
4.A.i                 z = lowest DFS-index neighbor;  newmsg.specialFlag = DEFAULT;
4.B.              else if (msg.source = dfsparent[y])
4.B.i.                z = glpn[y];  newmsg.specialFlag = PARENTFLAG;
4.C.              else if (msg.source = glpn[y])
4.C.i.                z = dfsparent[y];  newmsg.specialFlag = LOWPOINTFLAG;
4.C.ii.               newmsg.glpv = msg.sourcedfs;
4.D.              else if (msg.sourcedfs < dfs[y])
4.D.i.                z = glpn[y];  newmsg.specialFlag = PARENTFLAG;
4.E.              else //Comment: msg.sourcedfs < dfs[y]
4.E.i.                z = lowest DFS-index neighbor that is not the same as msg.source;
4.E.ii.               newmsg.specialFlag = DEFAULT;
5.            Send newmsg to node z.
```

Fig. 4. Rules to forward a SEARCH message when received by a node y that is not added to the trees yet

cycle, the message either reaches the drain or any other node that is already added to the trees (in CYCLE state) without backtracking. The specialFlag in the message is set to DEFAULT (refer to Step 4.A of Figure 4).

Case 2: $\mathtt{dfs}[\mathtt{x}] < \mathtt{dfs}[\mathtt{y}]$ *and* x *is the DFS-parent of* y.
In this case, if there exits a node z in the neighborhood of y whose DFS-index is lower than that of x, then the message could be forwarded to node z. If such a node does not exist, then node y forwards the message to its GLPN. The specialFlag in the message is set to PARENTFLAG indicating that the message was received from a DFS-parent, hence must be forwarded to the GLPN successively (refer to Step 4.B of Figure 4). The

message is forwarded along the generalized low-point path (refer to Step 2 of Figure 4). The node that forwards this message to the low-point node resets the flag to DEFAULT. From the low-point node onwards, the message is forwarded to the lowest DFS-index neighbor until it reaches the drain. Since the generalized low-point path does not involve any loops, a path is chosen for augmentation without backtracking.

The above two cases are sufficient if the colored trees are constructed to satisfy the CT-LD constraint. Hence, steps 1, 2, 4.A, 4.B, 4.E and 5 are sufficient in the set of rules to construct the colored trees satisfying CT-LD constraint. Note that if conditions 4.A and 4.B fail, then it implies that node x is the lowest DFS-index neighbor of y and is not its parent. Then, there must exist one node z in the neighborhood of y such that $\mathtt{dfs}[\mathtt{x}] < \mathtt{dfs}[\mathtt{z}] < \mathtt{dfs}[\mathtt{y}]$. One such obvious node is the DFS-parent of y. The message is forwarded along this neighbor with the DEFAULT flag. The message follows the lowest DFS-index neighbor successively to reach a node already added to the trees. The node at which the path augmentation terminates could be the same node that initiated the path search message, as the construction needs to satisfy the CT-LD constraint only.

However, if the colored tree construction were to satisfy the CT-ND constraint, then the nodes that start and terminate the paths must be distinct, for which the following rules are developed.

Case 3: $\mathtt{dfs}[\mathtt{x}] < \mathtt{dfs}[\mathtt{y}]$ *and* x *is the GLPN of* y.
In this case, dfs[x] is the GLPV of node y. The message in this case is forwarded to the DFS-parent of y and msg.specialFlag is set to LOWPOINTFLAG, indicating that the message was obtained from a low-point node, hence must be forwarded to the DFS-parent. In addition, the glpv field in the outgoing message is set to dfs[x] (obtained from the sourcedfs field of the received message, refer Step 4.C in Figure 4). Such a forwarding is continued until the message reaches a node whose GLPV is lower than msg.glpv, from where the message follows the generalized low-point path with the specialFlag in the message changed to PARENTFLAG (refer to Step 3 Fig. 4). From this point on, the forwarding of the message takes place similar to that of Case 2.

Note that when a message is forwarded to the DFS-parent upon msg.specialFlag = LOWPOINTFLAG, the message cannot reach the node that started the path search process, namely x. This would imply that the path search for augmentation started and ended at the same node. This in turn implies that no node in the DFS-tree beneath node x had a GLPV lower than dfs[x]. This contradicts the fact that when a network is two-vertex connected, then the GLPV of a child of x is strictly lower than dfs[x]. Hence, there exists an intermediate node whose GLPV is lower than that of dfs[x].

It can also be easily shown that the generalized low-point path taken from the intermediate node does not loop back to any of the nodes in the path through the DFS-parents as the intermediate nodes in the former path would have a GLPV value strictly lower than that at the intermediate nodes in the latter path. Hence, a path is chosen for augmentation without backtracking.

Case 4: $\mathtt{dfs}[\mathtt{x}] < \mathtt{dfs}[\mathtt{y}]$ *and* x *is neither the DFS-parent nor the GLPN of node* y.
In this case, node x is the lowest DFS-index node in the neighborhood of y and is not the GLPN of y. This implies that the GLPV of y is strictly lower than dfs[x].

The generalized low-point path of node y, by definition, does not contain x. Therefore, the message is forwarded to the GLPN of y with the specialFlag set to PARENTFLAG. The path taken by the message from then on is similar to that discussed in Case 2.

If the construction were to satisfy the CT-ND constraint, the forwarding algorithm will not reach Step 4.E as one of the earlier four cases would definitely hold true.

Every node attempts to find a path through each of its neighbor (except through the node from which it received the token), the number of SEARCH messages sent in the network is $2|\mathcal{L}| - (|\mathcal{N}| - 1)$. Every SEARCH message has a corresponding SUCCESS message. In addition, every node except the drain receives the TOKEN message to initiate a path search and sends a RETURN message. The total number of messages sent in the network is $8|\mathcal{L}| - 2|\mathcal{N}| + 2$. As the distributed algorithm is sequential in nature, the number of messages sent directly provides the running time of the distributed algorithm, which is linear in the number of links in the network.

Proof of correctness. The distributed algorithm is based on the path augmentation technique [17]. Hence, the proof of correctness of the algorithm follows from [17] and is not repeated in this paper due to space constraints.

5 Performance Evaluation

The linear-time distributed algorithm developed in this paper is evaluated on random topologies with 100, 200, 300, and 400 nodes. The topologies were constructed using Waxman's model [20]. The effectiveness of employing the generalized low-point (GLP) is studied by comparing the performance of the algorithm with that employing the traditional low-point (TLP) concepts. For each network size, twenty different topologies were simulated and the average results are shown in Table 1 for the CT-ND case. The "average minimum (maximum) path length" refers to the lowest (highest) path length among the two paths, averaged over all the nodes in the network. It is observed that a significant reduction in the average path lengths is obtained by employing the generalized low-point concept, which allows optimization of hop-count on the low-point path. The number of messages used in both the approaches were the same. Similar results were obtained for the CT-LD case and are not shown here due to space constraints.

Table 1. Comparison of the results of the distributed algorithm to compute colored trees satisfying CT-ND constraint employing traditional low-point and generalized low-point concepts

Number of Nodes	Average Number of Links	Average Red Path Length		Average Blue Path Length		Average Minimum Path Length		Average Maximum Path Length		Average Total Path Length		Reduction in Average Total Path Length
		TLP	GLP	TLP	GLP	TLP	GLP	TLP	GLP	TLP	GLP	
100	**774.2**	6.82	5.53	10.85	5.69	4.81	3.80	12.86	7.42	17.67	11.22	36.5%
200	**1382.65**	11.68	11.49	14.36	8.10	7.65	6.08	18.39	13.51	26.04	19.59	24.8%
300	**2585.56**	15.21	14.24	20.40	9.05	9.78	7.10	25.83	16.19	35.61	23.29	34.6%
400	**4540.45**	16.49	15.99	20.56	8.67	10.82	6.93	26.23	17.73	37.05	24.66	33.4%

6 Conclusions

This paper develops a linear-time distributed algorithm for the construction of colored trees for link/node-disjoint multipath routing to a particular drain in the network. The total number of messages sent in the network is shown to be $8|\mathcal{L}| - 2|\mathcal{N}| + 2$. The paper also demonstrates that significant reduction in the average path lengths may be obtained by employing generalized low-point concept in a DFS-tree rather than the traditional low-point concept.

References

1. Ye, Z., Krishnamurthy, S., Tripathi, S.: A framework for reliable routing in mobile adhoc networks. In: Proceedings of IEEE INFOCOM'03. (2003) 270–280
2. Pham, P.P., Perreau, S.: Performance analysis of reactive shortest path and multipath routing mechanism with load balance. In: Proceedings of IEEE INFOCOM. Volume 1. (2003) 251–259
3. Murthy, S., Garcia-Luna-Aceves, J.J.: Congestion-oriented shortest multipath routing. In: Proceedings of IEEE INFOCOM. Volume 3. (1996) 1028–1036
4. Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly resilient energy-efficient multipath routing in wireless sensor networks. ACK SIGMOBILE Mobile Computing and Communications Review **4**(5) (2001) 11–25
5. Bhandari, R.: Survivable Networks: Algorithms for Diverse Routing. Kluwer Academic Publishers (1999)
6. Grover, W.D.: Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking. Prentice Hall Publishers, New Jersey, USA (2003)
7. Begen, A.C., Altunbasak, Y., Ergun, O.: Multi-path selection for multiple description encoded video streaming. In: Proceedings of IEEE International Conference on Communications. Volume 3. (2003) 1583–1589
8. Lee, S., Gerla, M.: Split multipath routing with maximally disjoint paths in ad hoc networks. In: Proceedings of IEEE ICC. (2001) 3201–3205
9. Nasipuri, A., Das, S.R.: On-demand multipath routing for mobile ad hoc networks. In: Proceedings of IEEE International Conference on Computer Communications and Networks. (1999) 64–70
10. Wu, J.: An extended dynamic source routing scheme in ad hoc wireless networks. In: Proceedings of 35th Annual Hawaii International Conference on System Sciences. (2002) 3832–3838
11. Marina, M.K., Das, S.R.: On-demand multipath distance vector routing in ad hoc networks. In: Proceedings of IEEE ICNP. (2001) 14–23
12. Park, V.D., Corson, M.S.: A highly adaptive distributed routing algorithm for mobile wireless networks. In: Proceedings of IEEE INFOCOM. (1997) 1405–1413
13. Raju, J., Garcia-Luna-Aceves, J.J.: A new approach to on-demand loop-free multipath routing. In: Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN). (1999) 522–527
14. Valera, A., Seah, W.K.G., Rao, S.V.: Cooperative packet caching and shortest multipath in mobile adhoc networks. In: Proceedings of IEEE INFOCOM. (2003) 260–269
15. Lee, S., Gerla, M.: Aodv-br: Backup routing in ad hoc network. In: Proceedings of IEEE WCNC. (2000) 1311–1316
16. Ramasubramanian, S., Krishnamoorthy, H., Krunz, M.: Disjoint multipath routing using colored trees. Technical Report, University of Arizona (2005)

17. Medard, M., Barry, R., Finn, S., Gallager, R.: Redundant trees for preplanned recovery in arbitrary vertex- redundant or edge redundant graphs. IEEE/ACM Transactions on Networking **7**(5) (1999) 641–652
18. Xue, G., Chen, L., Thulasiraman, K.: Quality-of-service and quality-of-protection issues in preplanned recovery schemes using redundant trees. IEEE Journal on Selected Areas in Communication **21**(8) (2003) 1332–1345
19. Zhang, W., Xue, G., Tang, J., Thulasiraman, K.: Linear time construction of redundant trees for recovery schemes enhancing QoP and QoS. In: Proceedings of IEEE INFOCOM, Miami, FL, USA (2005) 2702–2710
20. Waxman, B.M.: Routing of multipoint connections. IEEE Journal of Selected Areas in Communications **6**(9) (1988) 1617–1622

Cross-Virtual Concatenation for Ethernet-over-SONET/SDH Networks*

Satyajeet S. Ahuja and Marwan Krunz

Dept. of Electrical and Computer Engineering,
The University of Arizona
{ahuja, krunz}@ece.arizona.edu

Abstract. Ethernet-over-SONET/SDH (EoS) with virtual concatenation is a popular approach for interconnecting geographically distant Ethernet segments using the SDH transport infrastructure. In this paper, we introduce a new concatenation technique, referred to as *cross-virtual concatenation* (CVC), which involves the concatenation of *virtual channels* (VCs) of hetrogenous capacities and can be implemented by a simple upgrade at SDH end nodes, thus utilizing the existing legacy SDH infrastructure. By employing CVC for EoS systems, we show that the SDH bandwidth can be harvested more efficiently than in conventional virtual concatenation. We later consider the routing problems associated with CVC connections, namely the connection establishment problem and the connection upgrade problem. We propose ILP and heuristic solutions to solve such problems. Simulations are conducted to evaluate the performance of the proposed heuristic and to demonstrate the advantages of employing CVC.

1 Introduction

Current optical transport infrastructure is dominated by the SDH technology [1]. SDH uses a bandwidth hierarchy indicated by STM-n, where $n = 1, 4, 16, 64, \ldots$. The basic unit in this hierarchy is the STM-1 channel (155.52 Mbps), which can support various smaller payloads, including VC-11 (1.5 Mbps), VC-12 (2 Mbps), and VC-3 (45 Mbps) channels. SDH was originally developed to support voice traffic. "Data" services are supported over SDH using Ethernet-over-SONET/SDH (EoS) with virtual concatenation [2].

In EoS with virtual concatenation, the aggregate bandwidth used for interconnecting two Ethernet segments is obtained by concatenating several SDH payloads (VC-n channels) of the same type, which can be independently routed to the destination. These channels, which we simply refer to as *virtual channels* (VCs), form a *virtually concatenated group* (VCG). Data is byte-interleaved over the various VCs of the VCG.

* This work was supported by NSF under grants ANI-0095626, ANI-0313234, and ANI-0325979, and by the Center for Low Power Electronics (CLPE) at the University of Arizona. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not reflect the views of NSF.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1039–1049, 2006.

Although the use of virtually concatenated EoS circuits has enabled efficient utilization of the SDH bandwidth, it may result in a large number of circuits between two end points. For example, consider two Ethernet LANs (with an average traffic of 100 Mbps between them) connected using a VCG of fifty VC-12 channels. Although such a connection greatly increases the bandwidth efficiency, the large number of circuits incur high maintenance overhead (the network management system has to maintain a large database to maintain these circuits). An alternative is to use three VC-3 channels (3×45 Mbps = 135 Mbps) at the expense of excessive bandwidth wastage. To achieve efficient bandwidth utilization using a relatively smaller number of connections, an EoS system needs to be able to combine VCs of different payloads in the same VCG. For example, the 100 Mbps traffic can be supported using two VC-3 channels (90 Mbps) plus five VC-12 channels (10 Mbps). We refer to such concatenation of SDH payload channels of different payload capacities as *cross-virtual concatenation* (CVC). In addition to reducing the maintenance overhead, CVC can also reduce the actual bandwidth usage. Due to hierarchical implementation of the SDH frame [1], only twenty one VC-12 (21 × 2Mbps = 42Mbps) channels or one VC-3 channel (45Mbps) can be transported over a TUG-3. This is because VCs typically incur some bandwidth overhead. For example, if we employ fifty VC-12 channels to support the EoS connection then it actually consuming 107 Mbps (45 + 45 + (8/21)×45) worth of SDH bandwidth. For the CVC case (two VC-3 channels plus five VC-12 channels), the actual SDH bandwidth consumed is 101.9 Mbps (2×45 + (5/21)×45).

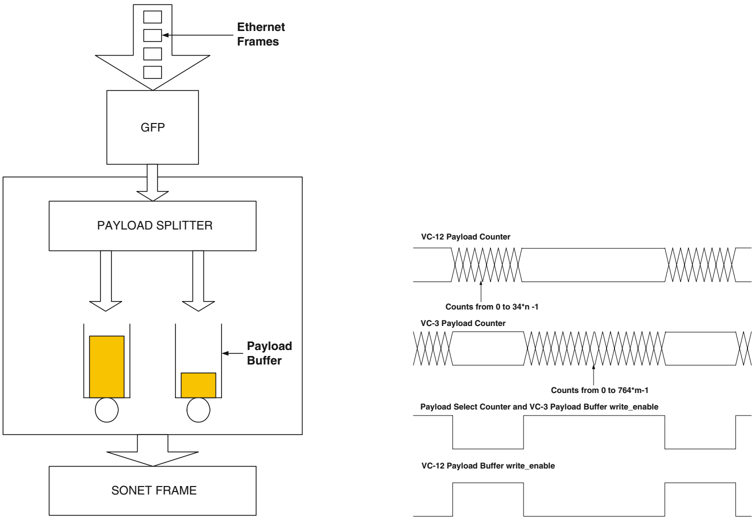
In this paper, we outline the general structure of the proposed CVC architecture and describe its advantages. We show a typical implementation of CVC using existing control overheads defined in SDH. The implementation requires a simple end-point upgrade. We study the path selection problems associated with CVC connection establishment. First, we consider the case of a new-connection establishment with a given bandwidth requirement. Then, we consider the case of bandwidth upgrade. We propose heuristic solutions for these problems. We study the performance of these heuristics and compare them with conventional VC case. Simulation results show that by employing CVC to establish EoS connections, we can harvest the SDH bandwidth efficiently.

2 Implementation of Cross-Virtual Concatenation

In this section, we describe how CVC can be implemented with a simple end-point upgrade. For illustration purposes, we describe a particular instance of CVC applied to the concatenation of VC-3 and VC-12 channels.

2.1 Transmit Side

In a typical transmit-side implementation of EoS, Ethernet frames are encapsulated using GFP (Generic Framing Procedure) [3].The resultant stream of bytes is then interleaved into various constituent members of the VCG. In the CVC implementation (see Figure 1(a)), we use a special payload splitter and a buffer assembly,



(a) Transmit side implementation of CVC.

(b) Timing diagram for the transmit side implementation of CVC.

Fig. 1. Transmit side implementation and timing diagram

just after packet encapsulation. The splitter is essentially a set of payload counters associated with each payload type. Each payload counter maintains the number of bytes of a particular payload in the SDH frame. For example, a VC-12 payload counter counts from 0 to $34n - 1$ for n VC-12 channels in the VCG. *payload-select* counter counts from 0 to $J - 1$, where J is the number of different types of payloads participating in CVC; in this case, $J = 2$. The various states of the payload-select counter are used to generate enable signals for various payload counters. At any instant, only one payload counter is enabled. There are J buffers (one for each payload type) to store the incoming payload bytes. When a byte is received after GFP encapsulation, it is stored into the payload buffer for which the payload buffer *write_enable* signal is high (see Figure 1(b)).

2.2 Receive Side

In a typical receive side implementation of EoS, the received SDH payload is stored in the differential delay compensator, which is essentially a payload buffer. Once the frames of all members of the VCG with the same multi-frame number (time-stamp) have arrived, the payload bytes are sequentially removed from the buffer based on the associated sequence number and are input to the GFP de-encapsulator. In a CVC implementation, we use different buffers for different payload types to compensate for the differential delay.

3 Connection Establishment Problem

Consider two inter-office Ethernet LANs that are to be connected using EoS with a given bandwidth requirement L. We define the *packing factor* α for the two payload types as the amount of bandwidth wastage incurred when using a smaller sized payload. For example, twenty one VC-12s consume the same SDH bandwidth as one VC-3, although the effective capacity of one VC-3 is 22.5 times that of a VC-12 channel. Hence, for these two payloads $\alpha = (22.5 - 21)/21$. In general, consider two payload types X and Y with bandwidth of B_x and B_y ($B_y > B_x$), respectively, and a packing factor α. Each link (i, j) is associated with two nonnegative integer capacity parameters C^x_{ij} and C^y_{ij} for the two payload types X and Y respectively. For a given source s and destination t, let $\mathbb{P}_x$ and $\mathbb{P}_y$ be k precomputed paths between s and t with capacity $f_x(p)$ ($\min_{(i,j)\in p} C^x_{ij} > 0$) and $f_y(p)(> 0)$ w.r.t. payload types X and Y, respectively. We now formally define the connection establishment problem.

Problem 1. (Connection Establishment): For a given graph $G(V, E)$, find a set of paths from s to t such that the total capacity of these paths is greater than L and the maximum differential delay between any two of them is less than a given constraint J. If there are multiple solutions, then find the one with the minimum bandwidth wastage.

In [4] the authors considered the so called Differential delay routing (DDR) problem and showed that it is NP-complete. The problem at hand is more complicated than the standard DDR problem, and in the best case is equivalent to DDR problem (when the two payload types are of the same capacity). Hence, the connection establishment problem is also NP-complete. This problem can be addressed by splitting it into two parts. First, we check if the requested bandwidth requirement is feasible or not (using standard maximum-flow algorithm [5]). Then, we find the set of paths that satisfy the differential delay requirement. This step requires enumeration of paths, and is therefore NP-hard. Instead of computing these paths at the time of connection establishment, we can precompute a set of K paths, for each of the payloads and then focus on choosing a set of paths that satisfy the differential delay and bandwidth requirement. We first propose an ILP formulation to solve such a problem and later propose an efficient heuristic based on the sliding-window algorithm.

3.1 ILP Formulation

Consider two sets of k paths $\mathbb{P}_x = \{P^x_1, \ldots, P^x_k\}$ and $\mathbb{P}_y = \{P^y_1, \ldots, P^y_k\}$ for the two payload types X and Y ($B_x < B_y$), respectively. Let x_{jk} and y_{jk} be the integer flow on on links (j, k) w.r.t. payload types X and Y, respectively. Let X_i and Y_i be the flow along path P^x_i and P^y_i. Constraint 1, 2, and 3 are the flow conservation constraints on nodes (see Figure 2). Constraint 4 is the capacity constraints on edges. Constraint 5 relate flows on edges to the flows on paths, and constraint 6 force integrality constraint on flows of payload types X and Y, respectively.

$$\text{Minimize} \sum_{\forall(j,k)\in E} x_{jk} + \sum_{\forall(j,k)\in E} y_{jk}$$

Subject to:

$$\sum_{k:(j,k)\in E} x_{jk} - \sum_{k:(k,j)\in E} x_{kj} = 0, \quad j \in V - \{s,t\} \tag{1}$$

$$\sum_{k:(j,k)\in E} y_{jk} - \sum_{k:(k,j)\in E} y_{kj} = 0, \quad j \in V - \{s,t\} \tag{2}$$

$$\left\{ \sum_{k:(j,k)\in E} (B_x x_{jk} + B_y y_{jk}) - \sum_{k:(k,j)\in E} (B_x x_{kj} + B_y y_{kj}) \right\} \begin{array}{l} \geq U,\ j = s \\ \leq -U,\ j = t \end{array} \tag{3}$$

$$0 \leq x_{jk} \leq C^x_{jk},\ 0 \leq y_{jk} \leq C^y_{jk}, \quad (j,k) \in E \tag{4}$$

$$x_{jk} = \sum_{i:(j,k)\in P_i} X_i, \quad y_{jk} = \sum_{i:(j,k)\in P_i} Y_i, \quad (j,k) \in E \tag{5}$$

$$X_i \in \{0,1,2,...,\lceil U/B_x \rceil\},\ Y_i \in \{0,1,2,...,\lfloor U/B_y \rfloor\}, i = 1,2,...,N \tag{6}$$

Fig. 2. ILP formulation for the connection establishment problem

3.2 Sliding Window Algorithm

The set of paths returned by the ILP will minimize the bandwidth wastage but worst-case complexity associated with this algorithm can be exponential. Hence, we now propose a computationally practical heuristic, which is a variant of the sliding-window algorithm [4]. The algorithm sequentially tries a set of precomputed paths to find a feasible solution. In case of multiple solutions, it attempts to return a solution with minimum bandwidth wastage. The heuristic uses K-shortest path algorithm [6] to find the set of precomputed paths. As shown in the pseudocode in Figure 3, the inputs to the algorithm are the graph $G(V,E)$, source s, destination t, bandwidth requirement L, and differential delay J. The algorithm precomputes two sets of k paths, $\mathbb{P}_x$ and $\mathbb{P}_y$, for payload types X and Y. Create $\mathbb{P} = \mathbb{P}_x \cup \mathbb{P}_y$, ordered according to their delay values. In the j^{th} iteration, the algorithm considers all the paths $P_j, ..., P_r$, where P_r is the path with highest delay such that $d_r - d_j \leq J$, where d_i is delay of path P_i. The algorithm routes maximum flow along them, starting with the paths of positive payload capacities w.r.t. Y and then w.r.t. X. If the total flow is greater than L, the algorithm stores the solution with minimum bandwidth wastage. The wastage associated with any solution is the wastage incurred due to the use of X type payload.

4 Connection Upgrade Problem

We now consider the connection upgrade (CU) problem. Given an established EoS connection, we want to *upgrade* it with additional L bits/sec without affecting the traffic. LCAS [7] can be modified to incorporate CVC and additional bandwidth can be harvested multiple types of channels. In [8] the CU problem

```
**Sliding_Window**$(G(V,E), s, t, w(.), C_x, C_y, B_x, B_y, L, J)$
Set $Flow_x = 0, Flow_y = 0$, $W = \infty$, $R = \emptyset$
Precompute $k$ paths $\mathbb{P}_x = \{P_{x1}, P_{x2}, ..., P_{xk}\}$ and $\mathbb{P}_y = \{P_{y1}, P_{y2}, ..., P_{yk}\}$
    for payload type X and Y respectively with increasing order of delays
$P = \mathbb{P}_x \cup \mathbb{P}_y$, sort $P$ based on increasing values of path delays
For $i$ = 1, 2,...,2$k$,
    Consider paths $P_i, ..., P_r$, s.t. $P_r$ is highest delay path satisfying $d_r - d_i \leq J$
    For $j = i, i+1, ..., r$,
        Route maximum flow w.r.t. $Y$ along path $P_j$, $Flow_y = Flow_y + f_{yj}$
        Route maximum flow w.r.t. $X$ along path $P_j$, $Flow_x = Flow_x + f_{xj}$
    If $Flow_x * B_x + Flow_y * B_y > L$, /*A feasible solution found */
        $W_i$ = Wastage$(Flow_x, Flow_y, B_x, B_y, L, \alpha)$
        If $W \geq W_i$,
            Store paths $R = \{P_i, P_{i+1}, ..., P_r\}$, Update $W = W_i$
    Remove all flow routed in the network, Reset $Flow_x = 0, Flow_y = 0$
Return $R$
**Function: Wastage**$(x, y, B_x, B_y, L, \alpha)$ /*For $B_y > B_x$*/
If $L \geq B_y y$,
    $L = L - B_y y$, Return $\lfloor \frac{L}{B_x} \rfloor \alpha$
else    /*$L < B_y y$*/
    $L = L - B_y \lfloor \frac{L}{B_y} \rfloor$
    If $L > B_x x$, Return $B_y - L$
    else Return $\lfloor \frac{L}{B_x} \rfloor \alpha$
```

Fig. 3. Pseudocode for the sliding-window algorithm

for the conventional virtually concatenated EoS system was studied by modelling it as a TSCP (two-sided constraint path) problem and was shown to be NP-complete in [9]. The problem was heuristically solved using MLW-KSP algorithm. CU problem is a generalization of TSCP and hence is also NP-Complete.

Problem 2. Connection Upgrade: Given a graph $G(V,E)$, source s and destination t with an EoS connection of m members in the VCG, with path delays $D_1, \ldots, D_m$. Let J be the maximum allowable differential delay. Given a required upgrade bandwidth L, find set of paths S between s and t that satisfy the following: (1)Total bandwidth that can flow along these paths is greater than L, (2) Delay associated with $s_i \in S$ satisfies differential delay constraint with all paths in S and with all existing members of VCG. Specifically, $|D_{s_j} - D_{s_k}| \leq J$, $\forall s_j, s_k \in S$ and $\max_{1 \leq i \leq m} |D_i - D_{s_k}| \leq J$, $\forall s_k \in S$ and (3) wastage associated with S is minimum among all possible solutions.

We now discuss a heuristic approach for finding the set S. The inputs to the upgrade algorithm are the graph $G(V,E)$, $(s-t)$ pair, capacity constraint L, two positive constraints C_1 and C_2 representing possible maximum and minimum delay of paths in solution set S, and the packing factor α. The algorithm precomputes a set $\mathbb{P}$ using the MLW-KSP algorithm [8]. We then use a modified version of the sliding-window approach discussed in Section 3 to find the feasible and the most optimal solution in terms of bandwidth wastage. In the jth iteration, the algorithm considers the P_j^{th} path in $\mathbb{P}$, routes the maximum flow along P_j, and finds the new values of C_1 and C_2 after incorporating P_j in the VCG. The algorithm then finds the path with the maximum capacity among the set of paths in $\mathbb{P}$, which is also feasible with respect to the new values of C_1 and

```
**Upgrade_Algo**$(G(V,E), s, t, w(), C_x, C_y, C_1, C_2, \alpha)$
$Flow_x = 0, Flow_y = 0$, $W = \infty$, $R = \emptyset$ and $S = \emptyset$
Precompute $\mathbb{P} = \mathbb{P}_x \cup \mathbb{P}_x$ using MLW-KSP
For $i = 1, 2, ..., 2k$,
    $A_1 = C_1, A_2 = C_2$
    Augment $f_y(P_i)$ and $f_x(P_i)$ flow along path $P_i$
    $Flow_y = f_y(P_i), Flow_x = f_x(P_i)$, Update $A_1$ and $A_2, \mathbb{G} = \mathbb{P} - \{P_i\}, R = \{P_i\}$
    Calculate capacity of each path in $\mathbb{G}$, remove infeasible and zero capacity paths.
    While (G not empty),
        Find path $P_r$ with maximum capacity
        Augment maximum flow along $P_r$, $Flow_y = Flow_y + f_y(P_r)$, $Flow_x = Flow_x + f_x(P_r)$
        $R = R \cup \{P_i\}$, Update $A_1$ and $A_2$
        Re-Calculate capacity of each path in $\mathbb{G}$, remove infeasible and zero capacity paths.
    If $Flow_x * B_x + Flow_y * B_y > L$, /*A feasible solution found */
        $W_i$ = Wastage$(Flow_x, Flow_y, B_x, B_y, L, \alpha)$
        If $W \geq W_i$,
            $S = R$, $W = W_i$
    Remove all flow routed in the network
    $Flow_x = 0, Flow_y = 0$
Return $S$
```

Fig. 4. Pseudocode for the connection upgrade algorithm

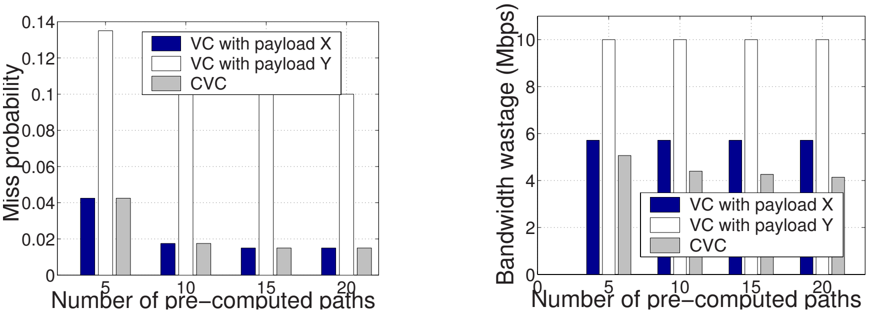
C_2. The algorithm continues to find the feasible path and route the flow until there are no feasible paths with positive flow. The pseudocode of the upgrade algorithm is presented in Figure 4.

5 Simulation Results

In this section, we use simulations to study the performance of the proposed algorithms. We report the gain achieved using CVC over the standard VC approach. Our simulations are based on random topologies that obey recently observed power laws [10] and are generated using the BRITE topology generator [11]. For a link (i, j), $f_x(i, j)$ and $f_y(i, j)$ are sampled from uniform distributions in the range $[0, 50]$ and $[1, 5]$, respectively. We randomize the selection of the s-t pair and fix the values of L and J for a simulation run. To model a meaningful EoS scenario, we choose the values of B_x and B_y such that they represent VC-12 and VC-3 SDH payload types. Specifically, we let $B_x = 2$ Mbps and $B_y = 45$ Mbps.

5.1 Results for the Connection Establishment Problem

We study the performance of the sliding-window algorithm proposed in Section 3 and compare it with standard VC. Our performance metrics are the bandwidth wastage and the probability of a miss. If the algorithm finds a set of paths that satisfy the required bandwidth, then we call it a *hit*; otherwise, it is a *miss*. For the standard VC case, we separately consider two types of payload and execute the sliding-window algorithm for each type. We assume all nodes are capable of using payload type Y by converting them into payload type X with a packing factor α. Figure 5(a) depicts the miss probability of various methods versus the number of precomputed paths used by the algorithm. The standard VC with payload type X and CVC perform significantly better than the standard VC with



(a) Miss probability vs. number of precomputed paths.

(b) Bandwidth wastage vs. number of precomputed paths.

Fig. 5. Comparison of CVC with sliding-window algorithm and standard VC with the two types of payloads ($L = 80$ Mbps, $J = 70$, and a network of 100 nodes)

payload type Y because each node has a cross-connect of granularity equivalent to payload type X. Hence every solution to a connection establishment problem that uses only payload type Y is a solution to the connection upgrade problem with CVC and in most cases is a solution to the standard virtual concatenation with payload type X. In Figure 5(b), we study the performance in terms of the average bandwidth wastage for various values of precomputed paths in the sliding-window algorithm. In the three considered methods, we count all the successful attempts to find the set of paths and average the bandwidth wastage associated with the solution. The performance of CVC with sliding-window algorithm improves when we consider more precomputed paths because of the larger solution space. Compared to standard VC techniques, the performance of CVC in terms of bandwidth wastage is significantly better and it further improves with the number of precomputed paths. The performance of the standard VC technique does not improve with the number of precomputed paths because

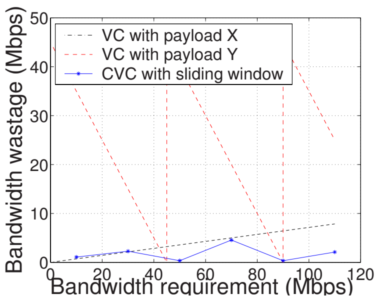
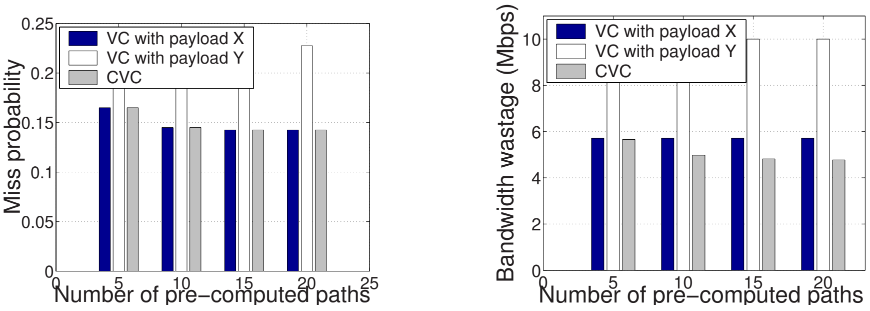


Fig. 6. Comparison between CVC with the sliding-window algorithm and the standard VC with the two types of payloads ($k = 15$, $J = 70$, and a network of 100 nodes)

for a given bandwidth and a standard VC technique, the amount of bandwidth wastage is fixed. To demonstrate the effectiveness of CVC, we plot the bandwidth wastage incurred by employing three types of concatenation techniques as a function of the traffic demand (L). As shown in Figure 6, the bandwidth wastage of the standard virtual concatenation with payload type Y follows a saw-tooth pattern, whereas it increases linearly in the standard VC with payload type X. CVC outperforms other techniques because it efficiently uses both payloads to achieve a lower bandwidth wastage. Recall that CVC has a significant advantage over standard VC with payload type X in terms of connection management (i.e., it requires fewer chennels).

5.2 Results for the Connection Upgrade Problem

For the connection upgrade problem, we study the performance of Upgrade algorithm proposed in Section 4. For a given simulation run, the values of C_1 and C_2 fall into the following cases: First, The shortest path between s and t w.r.t $w(.,.)$ is greater than C_2. In this case, there is no feasible solution. Second, The shortest path between s and t w.r.t $w(.,.)$ is less than or equal to C_1. The second case is nontrivial, and is the one considered in our simulations. Specifically, we let $C_1 = W(p*) + A + U(0, 50)$ ($U(x, y)$ is a uniform random variable with range (x, y)) and $C_2 = C_1 + J$, where $p*$ is the shortest path between s and t w.r.t. $w(.,.)$ and A is a positive constant. Figure 7(a) compares the performance of upgrade algorithm with the standard virtual concatenation in terms of miss probability for different values of precomputed paths used in the upgrade algorithm ($L = 80$ Mbps and $J = 70$). The performance of CVC is better than the performance of standard virtual concatenation with payload type Y and is same as the performance of payload type X with standard virtual concatenation. This is because the cross-connect chosen at each node assumes a granularity of B_x, and hence a



(a) Miss probability vs. number of precomputed paths.

(b) Bandwidth wastage vs. number of precomputed paths.

Fig. 7. Comparison of CVC with Upgrade algorithm and the standard VC with the two types of payloads, $L = 80$ Mbps, $J = 70$, and a network of 100 nodes

solution to the upgrade problem using standard virtual concatenation of payload type Y is also a solution of CVC and in most cases is the solution of standard virtual concatenation with payload type X. The standard virtual concatenation with payload type X and CVC performs significantly better than the payload type Y. The performance of CVC and standard concatenation with paylaod type X further improves by increasing the number of precomputed paths. In Figure 7(b), we study the performance of upgrade algorithm with the standard virtual concatenation in terms of average bandwidth wastage by varying the values of precomputed paths in the upgrade algorithm. The performance of upgrade algorithm improves when we increase the number of precomputed paths. Compared to the standard concatenation techniques the performance of CVC in terms of bandwidth wastage is significantly better and improves slightly when we use more number of precomputed paths in the upgrade algorithm. This is because with a larger solution space, the upgrade algorithm is able to efficiently allocate the resources within the two payload types such that the bandwidth wastage is minimized.

6 Conclusions

In this paper, we introduced the general structure of CVC (cross-virtual concatenation) for EoS circuits. We proposed a simple implementation of CVC that involves a simple upgrade at the end nodes and that reuses the existing SDH overhead. The algorithmic problems associated with connection establishment and connection upgrade were studied for CVC. We have proposed efficient heuristics to solve these problems using a sliding window approach. Extensive simulations were conducted to show the effectiveness of employing CVC for EoS systems.

References

1. ITU-T Standard G.707: Network node interface for the synchronous digital hierarchy (2000)
2. Ramamurti, V., Siwko, J., Young, G., Pepe, M.: Initial implementations of point-to-point Ethernet over SONET/SDH transport. IEEE Communications Magazine **42** (2004) 64–70
3. ITU-T Standard: G.7041 Generic Framing Procedure. (2003)
4. Srivastava, A., Acharya, S., Alicherry, M., Gupta, B., Risbood, P.: Differential delay aware routing for Ethernet over SONET/SDH. Proceedings of the IEEE INFOCOM Conference (2005, Miami)
5. Ahuja, R., Magnanti, T., Orlin, J.: Network flows: Theory, Algorithm, and Applications. Prentice Hall Inc. (1993)
6. Chong, E., Maddila, S., Morley, S.: On finding single-source single-destination shortest paths. Proceedings of the Seventh International Conference on Computing and Information (ICCI '95) (1995) 40–47
7. ITU-T Standard G.7042: Link capacity adjustment scheme for virtually concatenated signals. (2001)

8. Ahuja, S., Korkmaz, T., Krunz, M.: Minimizing the differential delay for virtually concatenated Ethernet over SONET systems. Proceedings of the IEEE 13th International Conference on Computer Communications and Networks, ICCCN **5** (2004) 205–210
9. Ahuja, S., Krunz, M., Korkmaz, T.: Optimal path selection for Ethernet over SONET under inaccurate link-state information. Proceedings of the Second International Conference on Broadband Networks (2005)
10. Faloutsos, M., Faloutsos, P., Faloutsos, C.: Power-laws of the Internet topology. Proceedings of the ACM SIGCOMM Conference (1999) 251–262
11. BRITE: Boston university representative Internet topology generator. (http://www.cs.bu.edu/brite/)

Optimal Wavelength Converter Placement with Guaranteed Wavelength Usage

Can Fang[1] and Chor ping Low[2]

[1] InfoComm Research Lab, ICIS, School of EEE, Nanyang Technological University,
639798, Singapore
fang0003@ntu.edu.sg
[2] ICIS, School of EEE, Nanyang Technological University,
639798, Singapore

Abstract. In this paper, we study the following problem. Given the network topology and traffic demand, determine how a minimum set of wavelength converters should be placed to ensure that the number of wavelengths needed will not exceed a given bound *L+u*, where *L* is the maximum link load in the network and u is a parameter defined by the network designer to reflect the overall availability of wavelength resources. This problem, however, is proved to be NP-hard. Hence we develop an efficient heuristic algorithm and extensive theoretical and experimental studies are carried out to verify the effectiveness and performance of the algorithm.

1 Introduction

Wavelength division multiplexing (WDM) [1][2] divides the bandwidth of an optical fibre into multiple wavelength channels so that multiple users can transmit data at distinct wavelengths through the same fibre concurrently. Since all-optical WDM networks can provide communication service with huge bandwidth and low latency, such networks are considered as candidates for the next generation wide-area networks which are required to meet the increasing traffic demand in the foreseeable future.

A *lightpath* is an optical communication path between a pair of source and destination nodes which may span multiple hops. In WDM networks, any pair of lightpaths (traffic demand) must be assigned with different wavelengths if they share the same link in any hops. Hence it is easy to see that the number of wavelengths required in a network is at least equal to the *natural congestion bound* or *maximum link load*, defined to be the maximum number of paths passing through any one link in the network.

Wavelength converter is an essential device in the multi-hop WDM networks that enhances the scalability of the network. In WDM networks without any wavelength conversion, the same wavelength must be assigned to all links in a lightpath (this is often referred to as the *wavelength continuity constraint*). If a node contains a wavelength converter bank, any lightpath that passes through this node may change its wavelength. Clearly wavelength assignments in networks with wavelength converters can be more efficient (uses less wavelengths) than wavelength assignments for the same set of paths where no wavelength converter is available. However, wavelength converters are

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1050 – 1061, 2006.

expensive devices and it has been anticipated that they will continue to be so in the foreseeable future [3]. In addition, densely placed converters may cause the signal distortion [4]. Hence, it is not practical to equip every node with a converter bank.

Several wavelength converter placement schemes [5][6][7] have been proposed in the literature to reduce the overall wavelength requirements of a given network by employing a minimal set of converters nodes. However, we note that existing converter placement schemes do not take into account the availability of resources, such as the number of wavelengths and converters that are available for utilization, in a given network; hence they are not able to adapt to the availability of resources of different networks.

In this paper, we aim to take into account above-mentioned issues into consideration in the design of efficient wavelength converter placement schemes for WDM networks. Furthermore, we aim to design a scheme that is able to provide a flexible trade-off between the number of wavelength converters to be placed and the number of wavelengths required to support the communications of all lightpaths in a given network. In particular, the problem that we interested in is, given the traffic demand in a network with arbitrary topology, locate a minimal set of converters nodes in the network such that the number of required wavelengths does not exceed a given upper bound $L+u$, where L is the maximum link load in the network and u is a parameter that can be defined by the network designer to reflect the overall availability of wavelength resources.

The rest parts of this paper are organized as follows: Section 2 presents the problem assumptions, formulation and the methodology used in this work. Section 3 addresses the problem of determining the wavelength requirements for the networks with special topologies. The results we obtained in Section 3 are applied in Section 4. In Section 4, a two-step algorithm is proposed and analyzed. Experimental study was carried out in Section 5. Section 6 concludes the paper.

2 Theoretical Preliminaries

2.1 Network Model

We model the network as a undirected simple graph $G\ (V,\ E)$, where V is the vertex set and E is the edge set. The traffic demand is represented by a set of lightpaths $D=\{l_1,\ l_2,\ l_3\ldots\ l_k\}$. In this paper, we consider the case of static routing where all connections (lightpaths) are known in advance and stay for an infinite period of time in the network. The number of wavelengths needed to support all lightpaths in D is denoted by $W(G,D)$.

We assume that all communications support *duplex communication channels,* whereby data can transmit in both directions in the same fibre. The set of lightpaths that occupy the same link must be assigned with different wavelengths on this link regardless of their transmitting direction.

In this paper, we assume all converters have *full conversion capability* [8][9], this means the converter can translate an incoming wavelength into any outgoing wavelength. We adopted the *shared by node* model [9], that is the converters placed at a node can be shared by any lightpaths that pass through this node.

Now we formally define the problem addressed in this paper as follows: given the network G and a set of traffic demand $D=(l_1,\ l_2,\ l_3\ldots\ l_k)$, locate a minimum set of nodes $S \subseteq V$ so that if we place wavelength converters at each node in S, the number of required wavelengths will not exceed the given bound L+u, where L is the maximum link load in the network and u is an integer parameter that can be defined by the network designer in the range of [*0, L/2*]. We refer this problem as *Optimal Wavelength Converter Placement with Bounded Wavelength Usage Problem* (*OPWB*).

2.2 The Computational Intractability of OPWB

Theorem 2.1 *OPWB* is NP-hard.

Proof. Suppose that *OPWB* is polynomial solvable, i.e., there is a polynomial algorithm $\mathcal{A}$ that can always yield an optimal solution $S \subseteq V$ for *OPWB*. Then it is apparent that $S \neq \phi$ if and only if $W(G,D)>L+u$. We therefore can determine whether $W(G,D)$ is larger than a given integer in polynomial time. Then by applying binary search, the exact value of $W(G,D)$ can also be calculated in polynomial time. However, to determine $W(G,D)$ for an arbitrary network G that with arbitrary traffic demand D is a NP-complete problem[10], so this algorithm $\mathcal{A}$ never exists and *OPWB* is NP-hard.

2.3 Graph Decomposition

Consider the case whereby a wavelength converter bank that is placed in a certain node v_i. All lightpaths that pass though v_i can convert their wavelength at v_i. The set of lightpaths that shared this converter are thus split into two parts, one from source node to the converter node v_i while another one from v_i to the destination node. The wavelength assignments for these two parts are independent from each other; thus placing a set of wavelength converters at a set of nodes S will result in the splitting of lightpaths that pass through the nodes in S into shorter lightpaths. This feature can be described by the *splitting operation* which is defined as follows:

Given a graph $G(V,E)$ and subset $S \subseteq V$, let $G_S(V',E')$ be a new graph derived from G by splitting each node $x \in S$ into $deg(x)$ one-degree nodes in V', where $deg(x)$ denote the degree of node x in G. Let $W_x \subseteq V'$ denote the set of vertices in G_S which are derived from node x in G. The process of decomposing node x in G into a new set of nodes W_x in G_S is referred to as the *splitting operation* (as in [6] & [7]). Fig 1 illustrates the decomposition of a given graph G into a new graph G_S by splitting nodes in the set S, where $S = \{3,4\}$. The process of having splitting operation on a graph G (V, E) can also be stated as: $G_S(V', E) = split\ [G\ (V, E), S]$.

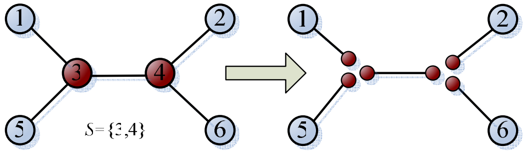


Fig. 1. Original graph G (V, E) and new graph $G_S(V', E)$ obtained by splitting operation

Since the task of wavelength assignment in networks with special topologies, such as paths, stars and trees, can be done more easily than in network with arbitrary topologies, we adopt the approach of decomposing a given network into edge-disjoint subgraphs with special topologies which include paths, stars and trees. The decomposition process is carried out by using the splitting operation described above. We note that such an approach has also been used in [6] and [7]. However the objectives of our approach differ from those in [6] and [7] as follow: the objectives of the work in [6]& [7] are to select a set of converters for placement to satisfy *L*-assignability and *3/2L*-assignability, respectively, i.e. fixed bounds on wavelength usage; on the other hand, the objectives of our approach is to place a minimal set of wavelength converters to satisfy *L+u*-assignability, where *u* is a parameter that may be specified by the user. Hence the problem addressed in this paper is a generalization of those addressed in [6] and [7].

3 Networks with Special Topologies

3.1 Network with Path Topology

Theorem 3.1 [6]. Given a network with path topology (which is often referred as *linear network*), denoted by G_{path}, then $W(G_{path}, D) = L$ holds for arbitrary *D*, where *L* is the maximum link load of G_{path}.

Theorem 3.2 [6]. Given a network *G*, if every connected component of *G* is a path, then $W(G, D) = L$ holds for arbitrary *D*, where *L* is the maximum link load of *G*.

It follows from Theorem 3.2 that if we split an arbitrary network into a set of linear networks, then *L*-assignability can always be achieved for the network. However, a major drawback of this approach is that a large number of nodes will have to be split in the process, thus resulting in high usage of wavelength converters.

3.2 Network with Star Topology

A *star* $G_{star}(V,E)$ is a graph whereby each vertex in G_{star} is of degree one except for one vertex whose degree is at least three. The vertex whose degree is three or above is referred to as *centre node* and all edges that adjacent to this vertex is called *legs*. We will show that the wavelength assignment problem for a network with star topology can be transformed to an *edge colouring problem* which is well studied.

Definition 3.1. Let *G* be a graph without loops, A *k-edge colouring* of *G* is an assignment of *k* colours to the edges of *G* in such a way that any two edges meeting at a common vertex are assigned with different colours. If *G* has a k-edge colouring, then *G* is said to be *k-edge colourable*. The chromatic index of *G*, denoted by $\chi'(G)$, is the smallest value of *k* for which *G* is *k*-edge colourable. The problem of finding a *k*-edge colouring of *G* whereby $k = \chi'(G)$ is called *edge colouring problem.*

Given a star network $G_{star}(V,E)$, we can construct a new graph $H^*(V^*,E^*)$ which we refer to as the *edge compatibility graph,* as follows.

Edge compatibility graph construction scheme (*EGCS*)

Input: A star network $G_{star}(V,E)$, $V=(v_1,v_2,...v_n)$, $E=(e_1,e_2,...e_m)$ and traffic demand $D=(l_1, l_2, l_3... l_k)$.

Output: Edge compatibility graph H^* ($V^*\cup W^*,E^*$).

1) $V^*=\phi$; $W^*=\phi$; $E^*=\phi$;
2) For each edge $e_i \in E$, create a vertex $v_i^* \in V^*$;
3) For each lightpath $l_i \in D$, we do the following:

   Case (*i*): l_i is a 2-hop lightpath.

   In this case, l_i will occupy two edges, say e_x and e_y in G_{star}. Insert an edge $e_i^* \in E^*$ in H^* that connects the two vertices $v_x^* \in V^*$ and $v_y^* \in V^*$ in H^* that correspond to the edges e_x and e_y.

   Case (*ii*): l_i is a 1-hop lightpath.

   In this case, l_i will occupy an edge, say e_x in G_{star}. Insert a new vertex $w_i^* \in W^*$ in H^* and insert an edge $e_i^* \in E^*$ in H^* that will connect the pair of vertices $v_x^* \in V^*$ and $w_i^* \in W^*$ in H^*.

Based on the construction scheme described above, it is easy to see that the *edge compatibility graph* H^* of a star network G_{star} satisfies the following properties:

- Each vertex $v_i^* \in V^*$ in H^* corresponds to an edge $e_i \in E$ in G_{star}.
- Each edge $e_i^* \in E^*$ in H^* corresponds to a lightpath $l_i \in D$ in G_{star}.
- Any two edges in H^* are adjacent if and only if their corresponding lightpaths occupy the same edge in G_{star}.

Since each pair of lightpaths in G_{star} must be assigned with different wavelengths if they occupy the same link, it is easy to see the task of assigning wavelengths to lightpaths in G_{star} is equivalent to that of assigning colours to the edges in H^* such that any two adjacent edges are assigned with different colours, i.e. solving the edge colouring problem on H^*. The edge colouring problem is known to be NP-hard [11][12] and various results have been proposed in the literature to provide upper bounds on the chromatic index of a given graph. Some of these results are listed as follow.

Bounds on the chromatic index:

König's Theorem [13]. If G is a bipartite multi graph whose maximum vertex degree is d, then its chromatic index $\chi'(G)=d$.

Shannon's Theorem [14]. If G is a multi graph whose maximum vertex degree is d, then $d \le \chi'(G) \le \frac{3}{2}d$ [14].

Vizing's Theorem (extended version) [15]. If G is a multi graph whose maximum vertex degree is d, and if h is the maximum number of edges joining a pair of vertices, then $d \le \chi'(G) \le d+h$.

Bounds on the wavelength requirement of a given network:

Theorem 3.3. Given a star network G_{star} with traffic demand D, it's maximum link load is denoted by L, let H^* be its edge compatibility graph constructed using *EGCS.* If H^* is a bipartite graph, then $W(G_{star}, D) = L$.

Proof. We note the maximum link load L of G_{star} is equal to the maximum degree d of H^*, thus it follows from König's theorem the chromatic index of H^* is equal to L. This in turn implies that the wavelength requirement of G_{star} is L.

Theorem 3.4. If G_{star} is a star network with traffic demand D, let h denote the maximum number of lightpaths occupying the same pair of edges (links) in G_{star}, and let L denote the maximum link load of G_{star} , then $W(G_{star}, D) \leq Min\ (3/2L, L+h)$ holds for arbitrary D.

Proof. Let H^* be the edge compatibility graph of G_{star} constructed by *EGCS*. The maximum link load L of G_{star} is equal to the maximum degree d of H^*. The maximum number of edges joining a pair of vertices in H^* is equal to the maximum number of lightpaths traversing the same pair of edges in G_{star}, i.e. h. Hence it follows from Shannon's Theorem and Vizing's Theorem that the chromatic index of H^* is bounded from above by $3/2L$ and $L+h$, respectively. This in turn implies that the wavelength requirement of G_{star} is bounded by $Min\ (3/2L, L+h)$.

3.3 Network with Bridges

Definition 3.2. Given a network $G\ (V, E)$, an edge $e \in E$ is called a *bridge* if $G - e$ is disconnected. Let C_1 and C_2 denote the two connected components of $G - e$, let $G_1 = C_1 \cup e$ and $G_2 = C_2 \cup e$, Then we say the two networks G_1 and G_2 are *singly connected* by bridge e.

Theorem 3.5. Given two networks G_1 and G_2, if G_1 and G_2 are singly connected by bridge e, then $W(G,D) = max[W(G_1,D_1), W(G_2,D_2)]$, where $G = G_1 \cup G_2$, $D = D_1 \cup D_2$; D_1 and D_2 is the set of lightpaths that traversing D_1 and D_2, separately.

Proof. Without lost the generality, we assume that $W\ (G_1, D_1) \geq W\ (G_2, D_2)$. We note that e is the only common edge of G_1 and G_2. Let T denote the set of lightpaths over e and $T = \{l_1, l_2, \ldots l_k\}$, $|T| = k$.

Consider the case whereby wavelengths have been assigned to all lighpaths in G_1 and G_2 using their respective assignment schemes, which we refer to as *Scheme 1* and *Scheme 2*.

We note that the wavelengths that have been assigned to G_1 and G_2 will form a *valid assignment* for G if the two schemes assign the same set of wavelengths to each lightpaths in T. The overall wavelength requirement of G in this case is $W(G,D) = W(G_1,D_1) = max[W(G_1,D_1), W(G_2,D_2)]$.

Next consider the case whereby Schemes 1 and 2 assign different set of wavelengths to the lightpaths in T. In this case conflict will arise between scheme 1 and scheme 2 in the assignment of the common lightpaths in T. In order to resolve this conflict, we can keep scheme 1 unchanged while reassigning the wavelengths in scheme 2 to satisfy:

i) All lightpaths in T will be assigned with same wavelength as in scheme 1;

ii) Any pair of lightpaths that are assigned with different wavelengths in scheme 2 before reassignment will still be assigned with different wavelengths.

This reassigning scheme is always possible to be carried out because scheme 1 uses no fewer wavelengths than scheme 2. Following the reassignment of wavelengths in G_2, the overall wavelength requirements of network G is again bounded by the $W(G_1,D_1) = max[W(G_1,D_1), W(G_2,D_2)]$.

Theorem 3.6. For a tree network G_{tree}, $W(G_{tree}, D) \le L+u$ if and only if for each star network $C_i \subseteq G_{tree}$, $W(C_i, D_i) \le L+u$, where D_i is the set of lightpaths that traversing C_i.

Proof. If: We note a tree $G_{tree}(V, E)$ can be constructed by taking a union of some connected components C_1, C_2, …, C_r, whereby the following conditions hold:

i). $G_{tree} = \bigcup_{i=1}^{r} C_i$;

ii). C_i is either a path or a star, for i =1,2, …, r;

iii). Given two components: $C_a = \bigcup_{i=1}^{m} C_i$, $C_b = C_{m+1}$, C_a and C_b are singly connected for $m=1,2,3,..r-1$.

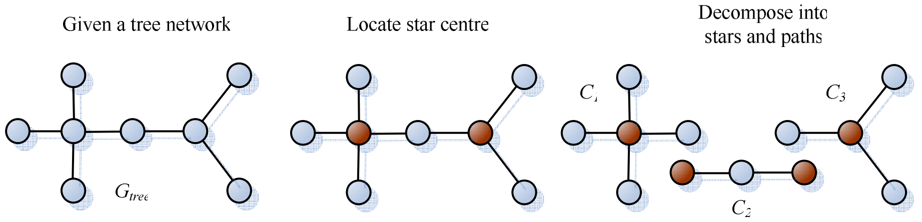


Fig. 2. Decompose a tree into a set of singly connected stars and paths

The process of decompose a tree $G_{tree}(V, E)$ into C_1, C_2, …, C_r is shown in Fig 2. Then from theorem 3.5 we have $W(G_{tree}, D) = Max[W(C_1, D_1), W(C_2, D_2), ..., W(C_r, D_r)]$, where $(C_1, C_2, ..., C_r)$ is a set of single connected star networks or linear networks that satisfy conditions i-iii stated above and $D_1, D_2, ..., D_r$ are the lightpath sets that traversing $C_1, C_2, ..., C_r$, separately. For each linear network C_j, theorem 3.2 shows that $W(C_j) = \le L \le L+u$, thus $W(G_{tree}, D) \le L+u$ holds if the wavelength usage of each star networks, $W(C_i, D_i)$, is bounded by $L+u$.

Only if: C_i is a sub-network of G_{tree}, so $W(C_i, D_i) \le W(G_{tree}, D)$, if $W(C_i, D_i) > L+u$, then we will have: $W(G_{tree}, D) \ge W(C_i, D_i) > L+u$, so $W(G_{tree}, D) \le L+u$ holds only when $W(C_i, D_i) \le L+u$.

4 Proposed Algorithm and Analysis

4.1 Algorithm for OPWB

As proved in [10], in general to determine the wavelength usage of a network is a NP-complete problem. In fact, to the best of our knowledge, no upper bound has been proposed for the wavelength usage of a network with arbitrary topology. In [7] Jia et. al showed that even for a network with simple topology (a 4-nodes graph), its wavelength requirement may exceed *3/2L*. Furthermore, in [16] Wilfong et. al showed that for a single ring network, its wavelength requirement may also exceed *3/2L*. Fortunately, if G is a tree network then its wavelength usage can be bounded by *3/2L* regardless of the traffic demand [17]. Based on this fact, in the first step of our algorithm we aim to determine the minimum set S_1 so that G_{S1} *(V', E) = Split* [*G (V, E), S_1*] will be a tree or a forest (a set of disconnected trees). This problem is often referred to as the minimum feedback set problem and is proved to be NP-complete [12]. However, as a well-studied problem, there exist many approximation algorithms with good performance guarantee. For example, in [18], a 2-approximate algorithm is proposed. Thus we can construct the vertex set S_1 which will be equipped with converters by applying these approximation algorithms.

After the converters are placed at each node in S_1, the wavelength usage of G_{S1} (*V'*, *E*) is bounded by *3/2L*. We can further tighten this bound by applying step 2. In this step, for each star $C_i \subseteq G_{S1}$, we examine the upper bound for its wavelength requirement determined by theorem 3.3 and theorem 3.4. For those star sub-networks whose upper bound exceed $L+u$, we include their centre nodes $v_i \in V'$ into set S_2. Converters will be placed at each node in S_2. After all these converters are placed, some stars are split into paths, the network G_{S1} is split into G_S and we can guarantee that for all remaining stars $C_i \subseteq G_S$, $W(C_i, D_i) \leq L+u$. Thus from theorem 3.6 we have $W(G_S, D) \leq L+u$, which implies that the total wavelength usage is bounded by $L+u$ and the total number of wavelength converters be placed is $|S_1| + |S_2|$.

Our algorithm can be described by the pseudocode:

Input: Network *G* (*V, E*), *V*={v_1, v_2...v_n}, *E*={e_1, e_2...e_m} with traffic demand set *D*=(l_1, l_2, l_3... l_k), the upper bound for the wavelength usage $L+u$.

Output: Vertex set *S*.

```
1)  Step one (place the converters at the feedback set nodes):
    $S = \phi, S_1 = \phi$ ;
    Find the minimum feedback set $S_1$ for $G$;
    $S = S \bigcup S_1$ ;
    $G_S(V,E)$ =split[$G(V,E)$,$S$];
2)  Step two (place the converters at the centre nodes of stars):
    $S_2 = \phi$ ;
    for $i$=1 to $n$ do:      /* for each vertex $v_i$ */
        if deg($v_i$) $\leq 2$     /*check the degree of vertices */
            $i$++;
        else                  /* $v_i$ is a center node of a star */
        build the edge-compatibility graph $H^*$ of the star
        with centre $v_i$ by the EGCS scheme
        check whether $H^*$ is a bipartite graph
        check the value of $l$ and $h$, which denotes the maxi-
        mum linkload and the maximum number of edges join-
        ing a pair of vertices, separately.
        if $H^*$ is not a bipartite graph and $Min(\frac{3}{2}l, l+h) > L+u$
        $S_2 = S_2 \bigcup v_i$ , $i$++;
        $S = S \bigcup S_2$ ;
    End and output vertex set $S$
```

4.2 Performance Analysis

(a) Computational complexity

Theorem 4.1. The computational complexity of proposed two-step algorithm is O ($|E||V|+|D||V|$).

Proof. In the first step, finding the minimum feedback set by the approximate algorithm proposed in [18] can be done in O ($|E||V|$) time; splitting operation can be done in O ($|E|$) time in the worst case. In the second step, each legs of the stars should be checked to determine the maximum link load of stars, we note each edge can be included in two stars at most so the complexity of checking link load is $O(|E|)$. Next we note the number of lightpaths after step 1 is $|D||V|$ at most, thus building edge compatibility graph by *EGCS* can be done in $O(|E|+|D||V|)$; checking whether the edge compatibility graph is bipartite for all stars can be done in O ($|E|+|D||V|$). Step 2 will cost O ($|E|+|D||V|$) and the two-step algorithm we proposed will cost O ($|E||V|+|D||V|$) in the worst case.

(b) The setting of u

As mentioned in section 1, the size of converter nodes set S is determined by network topology, traffic demand and given bound for the wavelength usage. In this section we will study the relationship between $|S|$ and the value of u:

1) $u=0$: In this case the wavelength usage is the minimum possible. Thus the size of S would be large. In the worst case, every star centre node of stars will be equipped with converter so the network will be split into a set of linear networks by S; this is the case that studied by [6].
2) $u=L/2$: It is proved in [17] and [7] that for the network with tree topology, this upper bound can always be meet for arbitrary traffic demand, thus we do not need place any converter in the second step. We can also note that in this case the *OPWB* is equal to the minimum feedback set problem, the wavelength converter placement problem under this case is studied by [7].
3) $0<u<L/2$, this is the general case that we are addressing in this paper, as shown our algorithm will generate a vertex set S with the size between case 1) and case2).

5 Experimental Study

In this section, we adopt experimental approach to study the relationship between the size of *S* and the value of *u*. The converter set *S* is constructed by the proposed algorithm. We assume the step 1 of the proposed algorithm has been done by the heuristics proposed in [18]. Three typical networks were studied which include NSFnet network, USA long haul network and mesh network. We also varied the size of mesh network from 4x4 to 7x7 to evaluate the effects of the network size. Some statistics of these networks are listed in Table 1.

Table 1. Statistics of some typical networks

Topology	Number of vertices	Feedback set size	Number of vertices whose degree larger than two
NSFnet	14	3	10
USA long haul	28	8	21
4×4 mesh	16	4	12
7×7 mesh	49	13	45

For each pair of vertices, they will generate a traffic demand at probability *p*, where *p* is a parameter controlling the total traffic load of the network. In this study we defined three types of traffic load condition:

i) Low traffic load, where *p* is set to 0.2;
ii) Moderate traffic load, where *p* is set to 0.5;
iii) High traffic load, where *p* is set to 0.8;

All traffic demands are routed by the shortest path algorithm. For each traffic load condition, we repeat the experiment by ten times to get the mean value of |*S*|, which denotes the size of wavelength converter nodes set. The results are shown in Fig 3-6.

The results clearly show that the size of *S* decrease about linearly as *u* increase from 0 to *L/2*, this relationship perfectly meet our expectation. With these |*S*|-*u* curves, the network designer can easily estimate the upper bound of wavelength usage when given the number of wavelength converters or estimate the number of required converters when given the upper bound for the wavelength usage.

Next we could note when the size of network increase, both the number of lightpaths and the maximum linkload increase. However, the |*S*|-*u* curve shows the same pattern, this means the size of network have little effect on the performance of the proposed algorithm.

We could also note when *u*=0, which means the wavelength usage is the minimum possible, the size of *S* constructed by the proposed algorithm is much smaller than the size of converter set constructed by the algorithm proposed in [6], which is equal to the number of vertices whose degree is larger than two (listed in the last column of Table 1). This result shows that taking the traffic demand into consideration will helps to reduce the redundant deployment of wavelength converters.

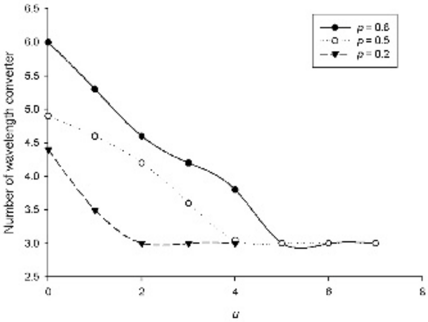


Fig. 3. Number of FWC in NSFnet

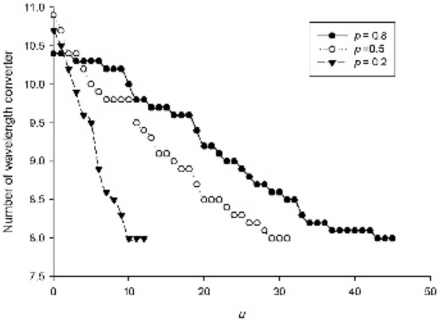


Fig. 4. Number of FWC in USA long haul network

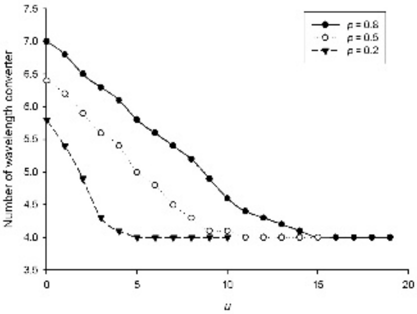


Fig. 5. Number of FWC in 4×4 Mesh network

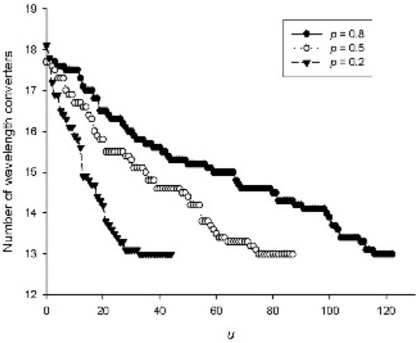


Fig. 6. Number of FWC in 7×7 Mesh network

6 Conclusion

We have studied the problem of placing a minimal set of wavelength converters in WDM networks with arbitrary topology and the total wavelength usage is bounded. The traffic demand is also taken into consideration. In this work, the network designer can set the upper bound for wavelength usage in the range of [*L, 3/2L*]. Thus the proposed algorithm is more flexible compared to existing work in this area. A two-step algorithm is proposed for this problem, its correctness is guaranteed by a set of theorems and its effectiveness is evaluated by theoretical and experimental studies.

This work can benefit WDM network design and development in several aspects. Firstly, by considering the traffic status, the number of converter can be further reduced compared to earlier works. Secondly, it can helps us to understand the relationship between the number of converters and the bound on wavelength usage, thus enable more efficient utilization of wavelength converters. Thirdly, by adopting our two-step algorithm and wavelength switching techniques, the wavelength assignment problem for a network with arbitrary topology can be reduced to a wavelength assignment problem in a set of independent stars and paths which in turn helps in reducing the overall computational complexity.

References

1. P.E. Green, Fiber-Optic Networks, Prentice-Hall, Cambridge, MA, 1992.
2. P.E. Green, Optical Networking update, IEEE journal on Selected Areas on Communication, vol.14, pp. 764-779, June 1996.
3. J.M.H Elmirghani, H.T. Mouftah, All-optical wavelength conversion technologies and applications in DWDM networks, IEEE Communications Magazine 38 (3) 2000, pp 86-92.
4. M. Attygalle, Y.J. Wen, A. Nirmalathas, Cascaded operation of all-optical wavelength converters based on nonlinear optical loop mirror, Lasers and Electro-Optics Society. LEOS 2002, Volume 2, 10-14 Nov. 2002, vol.2 pp 463 – 464.
5. J. Kleinberg and A. Kumar, Wavelength conversion in optical networks, Proc. 10th ACM-SIAM Symp. Discreate Algorithms (SODA), 1999, pp566-575.
6. X.H. Jia, D.Z. Du, X.D. Hu, H.J. Huang, D.Y. Li, On the optimal placement of wavelength converters in WDM networks, Computer Communications 26(2003), pp986-995.
7. X.H. Jia, D.Z. Du, X.D. Hu, H.J. Huang, D.Y. Li, Optimal placement of wavelength converters for guaranteed wavelength assignment in WDM networks, IEICE Trans on Communication, vol.E85-B, September, 2002, pp1731-1739.
8. R. Ramaswami and K. N. Sivarajan, Routing and wavelength assignment in all-optical networks, IEEE/ACM Trans. Networking , vol. 3, Oct 1995, pp489-500.
9. K. C. Lee and V. O. K. Li, A wavelength-convertible optical network, Journal on Lightwave Technology, vol.11, May/June 1993, pp962-970.
10. I. Chlamtac, A. Ganz and G. Karmi, Lightpath communications: an approach to high bandwidth optical WAN's, IEEE Transactions on Communications, vol.40, Issue 7, July 1992, pp1171 – 1182.
11. Marek Kubale, Graph colorings, Providence, R.I, American Mathematical Society, 2004.
12. M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, San Francisco, CA, 1979.
13. D. König, Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre, Math. Ann. 77,1916, pp453-465.
14. C. E. Shannon, A theorem on coloring the lines of a network, J. Math. Phys. 28, 1949, pp148-151.
15. V. G. Vizing, On an estimate of the chromatic class of a *p*-graph, Metody Diskret. Analiz. 3(1964), pp9-17.
16. G. Wilfong and P. Winkler, Ring routing and wavelength translation, Proc. 10th ACM-SIAM Symp. Discrete Algorithm (SODA), 1998, pp333-341.
17. P. Raghavan and E. Upfal, Efficient routing in all-optical networks, Proc. 26th Annual ACM Symp. Theory of Computing (STOC), 1994, pp134-143.
18. V. Bafna, P. Berman and T. Fujito, A 2-approximate algorithm for the undirected feedback set problem, SIAM Discrete Math, vol.12, no.3, 1999, pp289-297.

Estimating Network Offered Load for Optical Burst Switching Networks

Przemyslaw Lenkiewicz[2], Marek Hajduczenia[1,3]
Mário M. Freire[2], Henrique J.A. da Silva[3], and Paulo P. Monteiro[1,4]

[1] Siemens S. A., Research and Development Department,
Rua Irmãos Siemens, nº 1, 2720-093 Amadora, Portugal
[2] Department of Informatics, University of Beira Interior,
Rua Marquês d'Avila e Bolama, 6201-001 Covilhã, Portugal
[3] Faculdade de Ciências e Tecnologia,
Universidade de Coimbra - Pólo II, 3030-290 Coimbra , Portugal
[4] Instituto de Telecomunicações – Pólo de Aveiro,
Universidade de Aveiro, 3810-193 Aveiro, Portugal
przemek.lenkiewicz@gmail.com,
marek.hajduczenia@siemens.com, mario@di.ubi.pt,
hjas@ci.uc.pt, paulo.monteiro@siemens.com

Abstract. The Optical Burst Switching (OBS) technology itself is still in the early stage of development and various studies are often performed independently, resulting in difficult comparison between individual data sets (including such controversial studies as burst/packet loss evaluation). To facilitate the future research and evaluation of OBS networks, in this paper we examine relations between standard network parameters and the resulting network offered load, creating a common ground for comparing various simulation results. Means of estimating the resulting network offered load based on parameters describing the network topology and type of traffic are developed and examined for various simulation scenarios (topologies, loads, etc.). It is argued that, given the target offered load value for a given topology, it is always possible to estimate the required idle time which had to be applied in the network nodes, in order to keep the offered network load at the pre-defined level.

Keywords: Optical Burst Switching Networks, Network Simulation, Network Offered Load, Network Load Estimation.

1 Introduction

The growing interest on OBS networks and the increasing number of available simulation results, conducted using typically custom developed applications, result in a complete chaos when attempting to compare simulation results produced by independent researchers. The situation is further deteriorated by the lack of any common measures for even such basic values as effective and offered network loads, which are required for proper understanding of the operating conditions imposed on the network structure. Furthermore, many simulation results are produced disregarding the establishing of network operating conditions, thereby leading to irreproducible results, which can hardly be compared with any other research in this field. It is hereby

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1062 – 1073, 2006.

proposed to adhere to a very simple definition of offered network load, which is one of the prime measures of the OBS network operating conditions and, apart from network topology, link capacity, link length, etc., constitutes an important parameter when comparing various simulation results.

The remainder of this paper is organized as follows. Section 2 includes information about OBS networks and the research conditions. Section 0 presents a description of the simulation results, along with their analysis. Conclusions are presented in section 4, and are followed by literature references in Section 6.

2 Research Motivation and Methodology

2.1 Offered Load Versus Effective Load

The *capacity* of any data network can be defined as the amount of traffic that can be transferred through it in a unit of time. For example, assuming that a given network structure has four bi-directional links, each with a bandwidth of 1 Gbit/s, we might easily establish that the capacity of the network in question is 8 Gbit/s. On the other hand, the *network traffic load* is the amount of traffic that the users generate and try to transfer through the network, producing a certain amount of data transmission events, which require allocation of resources. The *network offered load* is therefore defined in a straightforward manner as the ratio between the total network traffic load and the network capacity, as indicated by equation (1). It should be noted that, according to this definition, the offered network load can be greater than 1, since users might generate more traffic than the maximum that the network structure can relay. Therefore, such a measure describes very well the network condition, since under light and moderate load conditions (where all or almost all generated traffic can be delivered without imposing packet loss) it will be characterised by a value smaller than 1, while network overflow (when users attempt to transmit more data than the given network structure can accept within a given time unit) results in an offered load value greater than 1. The network effective load is defined as the ratio between the carried traffic and the network capacity, as expressed by equation (2).

$$L_{offered} = \frac{\sum_{i=1}^{n} L_i}{\sum_{j=1}^{m} C_j} \tag{1}$$

$$L_{effective} = \frac{\sum_{i=1}^{n} M_i}{\sum_{j=1}^{m} C_j} \tag{2}$$

In these equations:

- $L_{offered}$ is the network offered load;
- $L_{effective}$ is the network effective load;
- L_i is the amount of traffic generated by a single user i (out of n) in a unit of time;

- *Cj* is the capacity of a single link *j* (out of *m*) in the network (here, for simplicity, we assume that all links have the same capacity);
- *Mi* is the amount of traffic carried over a given link *i*;
- *n* is the number of active users in the network (producing traffic);
- *m* is the number of links in the network.

Taking into consideration the previously examined network example with 8 Gbit/s of raw capacity, and assuming that active users generate 7 Gbit/s, the offered load of the network can be readily estimated as 0.875. It is therefore clearly visible that, contrary to the network effective load, offered load can significantly exceed 1, leading to congestion and packet loss.

2.2 Reconfigurable OBS Simulator

All the following simulation results were obtained using a custom-built, object oriented, event driven simulator of a generic OBS network, with in-built reconfiguration capabilities (based on text configuration files). In this simulator, all the physical components of the OBS network are represented as objects, and the events related to all network elements are processed by simulating the behaviour of said objects. By using a topology description file and standard input characteristics of traffic generation, we were therefore able to observe closely the network operation for a pre-defined period of time. In this way, we could obtain reliable statistics without building an actual OBS network test-bed. All the required system parameters can be set up either through a direct call to proper set-up procedures or through the aforementioned configuration files, including: the network topology, its parameters (user, node, and link properties), and the characteristics / shape / nature of the traffic produced by users. The network description is kept in a topology definition file. The results produced by the OBS simulator have been validated by comparison with those previously reported by other researchers, e.g. [1] . Additionally, prior to the development of this simulator, a number of tests were designed to check the behaviour of the software, once completed. These tests included simulation of various network topologies under varying load conditions, unbalanced load conditions, varying number of users, and irregular topologies. All the tests performed were completed successfully, thereby proving that the software was designed and developed correctly, reproducing all OBS network specific characteristics.

3 Research

3.1 Input Parameters

Selection of the OBS architectures to be used in the simulations was mainly driven by the requirement of testing various topologies, featuring different number of nodes, links, and throughput. Specifically, the following architectures were included in the evaluations of the proposed model:

- D2T: ring network with 16 nodes (Fig. 1.a);
- D3T(1,15,3) and D3T(1,15,5): chordal rings with chord lengths of 3 and 5 (Fig. 1.b and c);

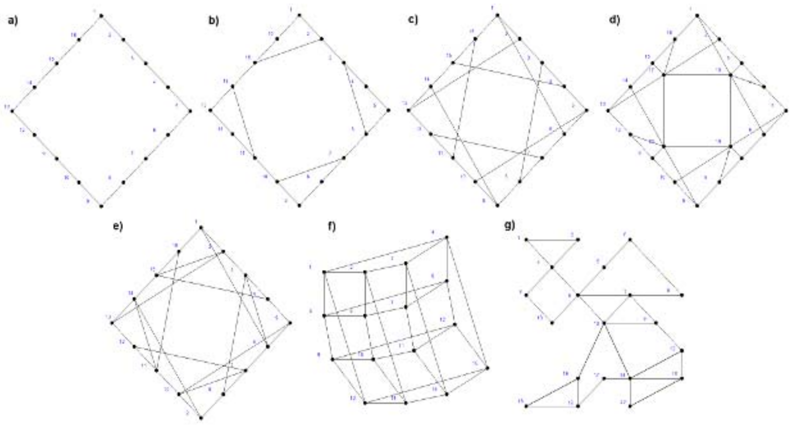


Fig. 1. Network topologies included in the research

- D3T(1,15,5): chordal ring with 4 additional core nodes (Fig. 1.d);
- D4T(1,15,3,5): chordal ring with 2 chords (lengths of 3 and 5) (Fig. 1.e);
- MeshTorus 16: mesh-torus network with 16 nodes (Fig. 1.f);
- MeshTorus 25: mesh-torus network with 25 nodes;
- Improvisation: randomly placed 20 nodes with 29 links (Fig. 1.g);
- GEANT: representation of GEANT network (http://www.geant.net);
- Very Simple: ring network with 4 nodes.

To simplify the examination of the resulting data sets, all links were assumed to have the same propagation delay and the same bandwidth. The Dijkstra [2,3] algorithm was used for control packet routing within the OBS network structure. The input traffic fed into the OBS network structure was produced using the standard Poisson random number distribution, and it is typically described using two parameters, namely the *average burst length* and the *average node idle time*. The former parameter describes the length of the burst expressed in a common reference time unit (milliseconds in this case, which are converted into picoseconds, selected as a common time reference base for the whole simulation), while the latter one expresses the length of the idle cycle for the given node between generations of two bursts (again expressed in milliseconds, converted into picoseconds for system compatibility). In terms of actual network operation, the aforementioned parameters express the traffic creation intensity. The former one describes the amount of time it takes for a given burst to be transferred from source to destination, while the latter one describes the amount of time it takes for a node to collect sufficient number of packets to meet its burst assembly conditions (depending on the employed burst aggregation scenario).

Each complete simulation scenario included 50 consecutive simulations using the same set of input network parameters. Once completed, the network parameters were altered accordingly, and the simulation cycle was repeated. The average burst size was varied between 150 and 10 milliseconds, with size values decreasing by 20% of their previous values in each simulation step. Similarly, the average node idle time was varied time between 20 and 1 millisecond, also decreasing by 20% in each simulation step. In total, 13 simulation steps were performed, producing network offered loads ranging from 0.02 to 46, thus reflecting all possible operating conditions for the OBS network (ultra light, light, moderate, heavy load, and network overload). Further increase/decrease in the network load would not contribute to more detailed analysis of the examined problem, and thus was avoided.

3.2 Estimation of Basic Approximation Curves

The estimated network offered load values obtained varied between 0.02 and 46, which represent two extreme network load conditions, when the OBS structure is very lightly loaded (only 2% of resources used) or flooded with data (46 times the nominal raw network capacity), as shown in Fig 2 (a) and (b). These figures depict the relation between the input simulation parameters, namely average idle time and average burst size, and the resulting offered load. It was observed that it is possible to express the network offered load, with reasonable accuracy, in terms of a power function of the average node idle time and in terms of a linear function of the average burst size. Since these dependencies constitute the grounds for the optimization study presented further on, it was decided to use those simple functions.

Following the aforementioned assumption about mixed power and linear function approximations of the network offered load, let us assume that the network offered load for a particular topology is described by equation (3), which depends only on the average node idle time (6). In order to provide the targeted relation between the network offered load and the pair average node idle time / average burst size (6), a more generic approximation (4) must be produced. Since (3) depends only on the average node idle time, the approximation coefficients p_1 and p_2 must be expressed in terms of average burst size by equations (5), where g and h are some generic functions, undefined at this point. In order to fully examine the relations between the network offered load and the pair average burst size / node idle time, several OBS network topologies were examined, with Fig. 2 depicting the obtained offered network load surfaces for a ring network with 16 nodes (Fig. 1.d) and a mesh-torus network with 16 nodes (Fig. 1.f).

$$L_{offered} = p_1 \cdot \hat{T}_{node}{}^{p_2} \tag{3}$$

$$L_{offered} = f\left(\hat{T}_{burst}, \hat{T}_{node}\right) \tag{4}$$

$$p_1 = g\left(\hat{T}_{burst}\right),\ p_2 = h\left(\hat{T}_{burst}\right) \tag{5}$$

$$\hat{T}_{burst},\ \hat{T}_{node} \tag{6}$$

Each selected network topology was further examined by performing a power function approximation of the measured network offered load values, following equation (3), thereby producing approximation curves similar to those presented in Fig. 3 (only two examples are depicted due to space limitation). Next, the data series were subject to power function regression, producing values of the targeted approximation coefficients p_1 and p_2, which were later on collected for each particular topology and depicted against average burst time, as shown in Fig. 4. Fig. 4.(a) presents the relation between the network topology, average burst size,

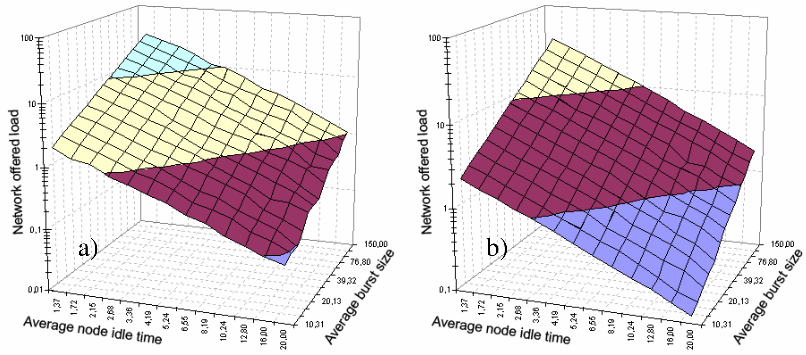


Fig. 2. Relation between network offered load and input traffic parameters in log scale for: (a) a ring network with 16 nodes (Fig. 1.d) and (b) a mesh-torus network with 16 nodes (Fig. 1.f)

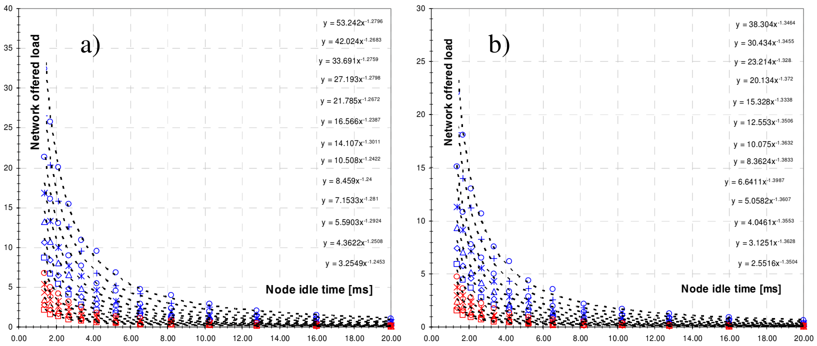


Fig. 3. Network offered load approximation against average node idle time (power function regression (3)) for: (a) a ring network with 16 nodes (Fig. 1.d) and (b) a mesh-torus network with 16 nodes (Fig. 1.f)

and approximation coefficient p_1 in equation (3), while Fig. 4.b presents the corresponding relation though this time for the coefficient p_2 in equation (3). It is visible that the coefficient p_1 exhibits a power function character when plotted against average burst size, while the coefficient p_2 has a quasi linear (constant) value with tiny fluctuations around the average value.

Next, when examining the relations in Fig. 4, it is immediately visible that the approximation coefficients p1 and p2 exhibit significant correlation with the network topology, which was expected, when comparing the examples of the network offered load surfaces for two different topologies, depicted in Fig. 2. Since the main goal of this paper is to produce a topology-independent, generic formula describing the offered network load as a function of the average node idle time and burst size (both expressed in milliseconds), it is necessary to establish a relation between each obtained topology-dependent curve and a particular examined OBS network topology.

First, the resulting relations (depicted in Fig. 4) had to be described as a function of the average burst length, by using a power function regression in the case of the p1 coefficient (7) and a strictly linear function regression in the case of the p2 coefficient (8). Then, the obtained regression coefficients had to be described as a function of the examined OBS network topology. Since there is no observable relation between the regression coefficients and any straightforward network parameters (number of nodes, edge nodes, links etc), two new network measures had to be devised and examined, namely:

- *node-link density*, which is hereby defined as the total number of nodes in the given topology divided by the number of links interconnecting the said nodes – equation (9), where N_{node} and N_{link} are numbers of nodes and links, respectively;
- *network diversity*, which is hereby defined as the difference between the total number of links in the network and the network diameter – equation (10), where D is the network diameter;

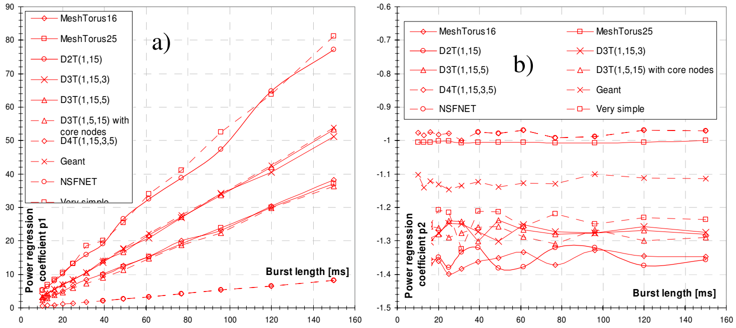


Fig. 4. Established relations between: (a) power approximation coefficient p_1 and (b) power approximation coefficient p_2 as a function of average burst length, for examined topologies

$$p_1 = p_1' \cdot \left(\hat{T}_{burst}\right)^{p_1''} \tag{7}$$

$$p_2 = p_2' \cdot \left(\hat{T}_{burst}\right) + p_2'' \tag{8}$$

$$\rho_{node/link} = \frac{N_{node}}{N_{link}} \tag{9}$$

$$\vartheta = N_{link} - D \tag{10}$$

$$p_1',\ p_1'' \tag{11}$$

$$p_2',\ p_2'' \tag{12}$$

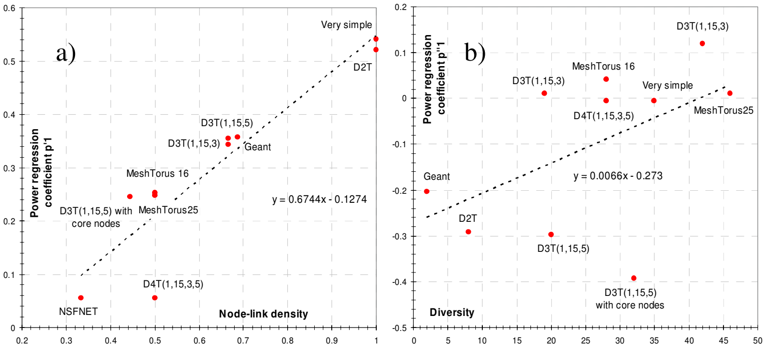


Fig. 5. Power function regression coefficients (11) (Fig. 4.a) and their relation with (a) node-link density (9), for the first regression coefficient, and (b) network diversity (10), for the second regression coefficient

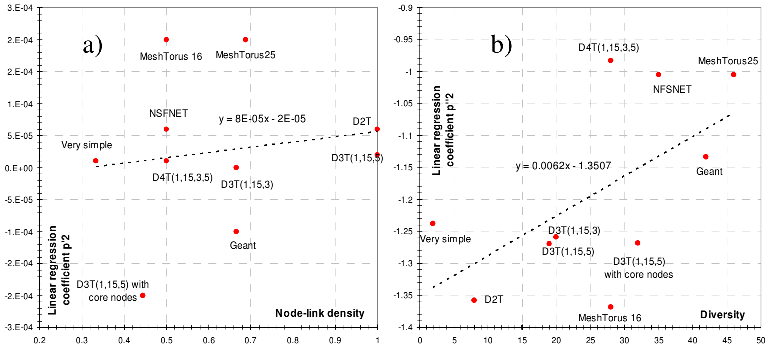


Fig. 6. Linear function regression coefficients (12) (Fig. 4.b), and their relation with (a) node-link density (9), for the first regression coefficient, and (b) network diversity (10), for the second regression coefficient. The first coefficient was disregarded (< 10^{-4}).

It was observed that the first of the regression coefficients (11) follows the above-defined node-link density coefficient (9) in a strictly linear manner, showing virtually no discrepancy for the examined topologies (see Fig. 5.(a)). The second of the regression coefficients (11) proved to be virtually topology independent and maintain a quasi-constant value as a function of the network diversity parameter (10) (see Fig. 5.(b)). In the case of the p_2 coefficient, the linear regression equation (8) proved to be virtually independent of the average burst size (see Fig. 4.(b)), and thus the first of the linear regression coefficients (12) is expected to have a value close to 0, which is further confirmed by Fig. 6.(a), where the values of this regression coefficient have a magnitude smaller than 10^{-4}. Further considerations will therefore omit this parameter due to its marginal value. The second of the regression coefficients (12) was plotted as a function of the network diversity parameter (10) (Fig. 6.b), and exhibits a linear dependence, though this time with significant discrepancy from the approximated value.

The final equations that model the relation between the network topology description and the initial power law regression coefficients for network offered load are (17) and (18), while (13) to (16) are partial equations obtained from the respective regressions. It is worth noting that only p_1 depends on the average burst size, while p_2 seems to depend only on the network diversity parameter.

$$p_1' = 0.6744 \cdot \rho_{node/link} - 0.1274 \tag{13}$$

$$p_1'' = 0.066 \cdot \vartheta - 0.273 \tag{14}$$

$$p_2' = 0 \tag{15}$$

$$p_2'' = 0.0062 \cdot \vartheta - 1.3507 \tag{16}$$

$$p_1 = (0.6744 \cdot \rho_{node/link} - 0.1274) \cdot \left(\hat{T}_{burst}\right)^{0.066 \cdot \vartheta - 0.273} \tag{17}$$

$$p_2 = 0.0062 \cdot \vartheta - 1.3507 \tag{18}$$

$$L_{offered} = \left\{ \begin{array}{l} (0.6744 \cdot \rho_{node/link} - 0.1274) \cdot \left(\hat{T}_{burst}\right)^{0.066 \cdot \vartheta - 0.273} \cdot \\ \cdot \left(\hat{T}_{node}\right)^{0.0062 \cdot \vartheta - 1.3507} \end{array} \right\} \tag{19}$$

The final expression linking the offered network load, average burst time, and average node idle time, is therefore (19), where the node link density and network diversity parameters describe each particular OBS network topology in a unique manner.

3.3 Result Validation / Precision

Once the final equation (19) linking the offered network load, average burst time, and average node idle time was established, its approximation accuracy required evaluation. The validation process consisted of calculating the network offered load surfaces (similar to those depicted in Fig. 2), comparing them with the simulation results, and presenting the resulting discrepancy (if any) between the calculated and simulated network offered load for all types of examined OBS topologies. The simulation process comprised 100 cycles per each data point (a pair of average burst size and node idle times), resulting in approximately 17,000 simulations per single examined topology, thereby providing a sufficient sample size to calculate and examine confidence intervals.

First, the D3T(1,15,5) network depicted in Fig. 1.d was examined, by producing the average network offered load surface depicted in Fig. 7.a. Fig. 7.b presents the size of the 95% confidence interval, proving that the obtained simulation results were consistent and repetitive. Fig. 8.a depicts the calculated network offered load surface, based on (19), with the discrepancy between the calculated and measured network offered load shown in Fig. 8.b.

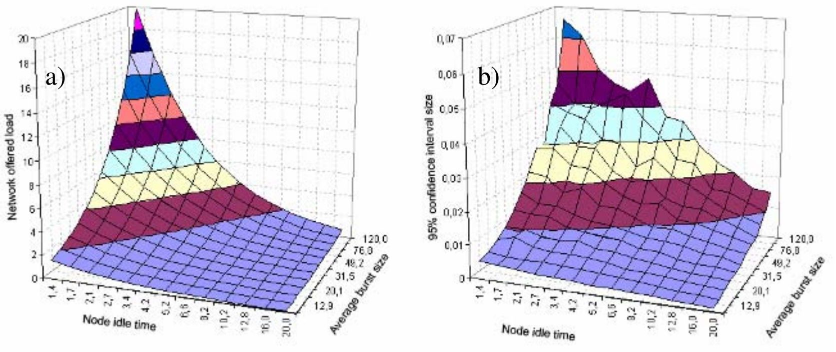


Fig. 7. (a) Simulated average network offered load for D3T(1,15,5) network (averaged over 100 simulations per data point) and (b) Size of the 95% confidence interval

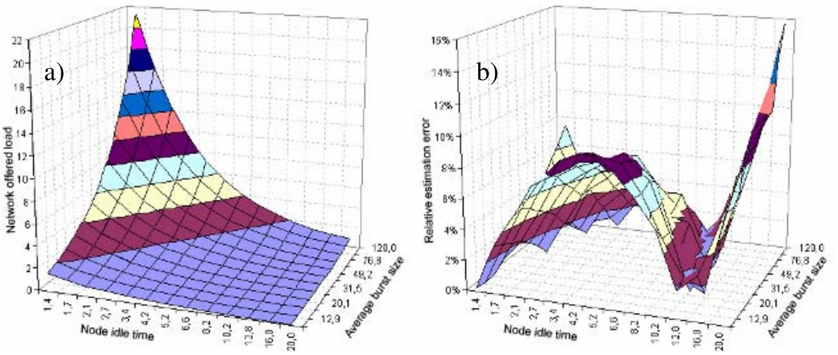


Fig. 8. (a) Calculated network offered load for D3T(1,15,5) network and (b) Relative difference between the simulated and calculated network offered load

It is a straightforward observation that the calculated network offered load surface is consistent with the simulated one, apart from the high overload area, where the estimation error is significant and exceeds 10%. However, it must be noted that a standard network is typically never subject to such high loads, exceeding the raw nominal capacity more than 20 times. The area of interest (offered network load ranging from 0 to 5) is approximated with very good quality, exhibiting estimation errors below 8%. Additionally, it must be noted that the observed estimation errors for very low offered loads (below 0.02 for very long average burst length, above 100 ms) results from finite simulation length. It was additionally observed that the said error diminishes as the simulation cycle length is increased, though that resulted in excessive simulation process time (in excess of 5 days for single average node idle time) and therefore was not further explored.

Next, a different network type was examined, namely a Mesh-Torus network with 16 nodes, as depicted in Fig. 1.f. The same simulation process conditions were applied also in this case, producing a complete set of characteristics. The average

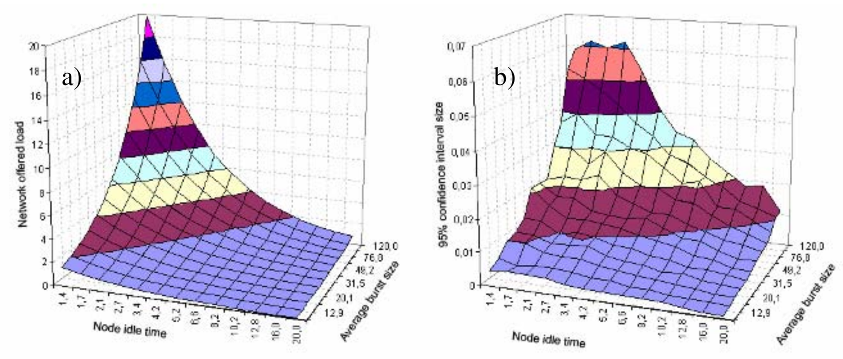


Fig. 9. (a) Simulated average network offered load for a Mesh-Torus network with 16 nodes (averaged over 100 simulations per data point) and (b) Size of the 95% confidence interval

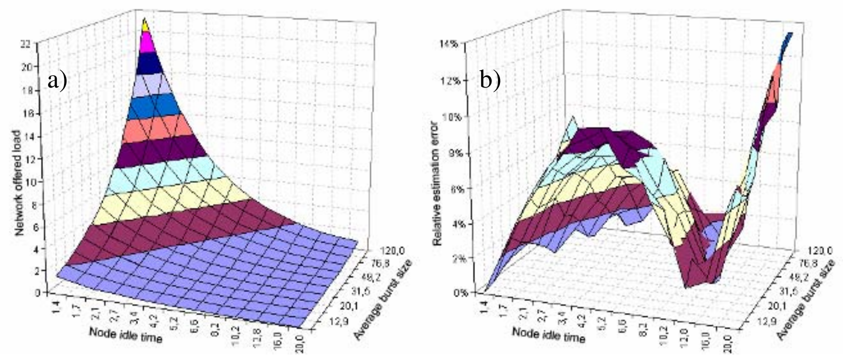


Fig. 10. (a) Calculated network offered load for a Mesh-Torus network with 16 nodes and (b) Relative difference between the simulated and calculated network offered load

network offered load surface is depicted in Fig. 9.(a), while Fig. 9.(b) shows the size of the 95% confidence interval, proving that the obtained simulation results were consistent and repetitive.

Fig. 10.(a) depicts the calculated network offered load surface, based on (19), with the discrepancy between the calculated and simulated network offered loads shown in Fig. 10.(b). Again, it is possible to observe that the calculated network offered load surface is consistent with the simulation results, with the mean estimation error below 8%. Slightly higher discrepancies are observed for very low network offered load with large average burst size (above 100 ms), the reasons for which were explained before. It is therefore concluded that the derived generic network offered load estimation equation (19), depending only on the generalized network topology description and input parameters such as average burst size and average node idle time, is consistent with the simulation results and can be used successfully to estimate the input values for said parameters, prior to performing any simulations.

4 Conclusions and Future Work

In this paper, we present a straightforward method to estimate the resulting network offered load based on such OBS network simulator parameters as network topology type and input traffic parameters (average node idle time and average burst size). The accuracy of this method was validated by comparing measured and calculated network offered load surfaces, producing estimation errors below 8% for moderate and high load while estimation error in excess of 15% was noticed only for ultra low network load with very large average burst size (in excess of 100 ms), where it is argued that increased simulation time leads to better convergence though extends the simulation process exorbitantly. However, since network offered load generalization is possible, assuming that a simple parameter (namely network topology name and description) is provided, the results of the corresponding simulations using various custom-built OBS simulators can have a common ground for comparison, once the network offered load matches. Additionally, using the approximation model we present in this paper, it is always possible to define the network operation point (average burst size and node idle time), and thus have a priori settings for the input traffic generator.

There is also room for future work, including improving the approximation precision of the general, topology-independent equation, and searching for new ways to describe differences between individual network topologies. Moreover, our study assumed that all links in the network share the same raw bandwidth (data rate and number of channels), which does not necessarily need to hold true in the case of real OBS networks. It is therefore imperative to evaluate the impact of varied link capacity on the network offered load. Other research issues for this topic include also OBS signalling algorithms, varied propagation delays (link lengths) and varied burst/packet loss probabilities.

Acknowledgments

This work has been financially supported by **Fundação para a Ciência e a Tecnologia (FCT)**, **Portugal**, through the grant contract **SFRH/BDE/15524/2004** and through **CONDENSA Project contract POSC/EEA-CPS/60247/2004** and by **Siemens S.A. COM RD1 R, Portugal**.

References

1. J. J. C. P. Rodrigues, M. Freire, and P. Lorenz: Performance Assessment of Signaling Protocols with One-Way Reservation Schemes for Optical Burst Switching Networks. High Speed Networks and Multimedia Communications (2004)
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein: Introduction to Algorithms, Section 24.3: Dijkstra's algorithm. pp.595-601, 2nd ed: MIT Press and McGraw-Hill, (2001)
3. E. W. Dijkstra: A note on two problems in connection with graphs, Numerische Mathematik. (1959) vol. 1, pp. 269 - 271

An Adaptive Parameter Deflection Routing to Resolve Contentions in OBS Networks

Keping Long[1], Xiaolong Yang[1,2,*], Sheng Huang[2], Qianbin Chen[2], and Ruyan Wang[2]

[1] Research Centre for Optical Internet and Mobile Information Networks (COIMIN),
University of Electronic Science and Technology of China, Chengdu 610054, China
Tel: +86-28-8320-7895; Fax: +86-28-8320-7885
yxl@uestc.edu.cn
[2] Chongqing Univ. of Posts and Telecomm., Chongqing 400065, China

Abstract. Currently, the contention resolution is one of research focuses for optical burst switching (OBS). The paper presents a new contention resolution scheme, named as ***adaptive parameter-based deflection routing***, which can control the deflection according to the time-varying traffic load and the QoS requirements. Compared with other schemes, the simulation results show that it can improve the overall *BLP* and the individual *BLP* of each class burst, and alleviate the offset-time deficit on QoS guarantee.

Keywords: Contention, Deflection Routing, Optical Burst Switching, QoS.

1 Introduction

Currently, many approaches[1]-[5] are proposed to resolve the burst contentions for OBS networks. Among them, the deflection routing is much more promising because of its lower requirements for optical components. However, the existing deflection routing algorithms[3]-[5] have some drawbacks in the control strategy and the path optimization. Motivated by the situations, the paper proposes a new contention resolution scheme, called ***Adaptive Parameter Deflection Routing*** (***APDR*** for short), which features largely in the adaptivity to the traffic load and the QoS requirements.

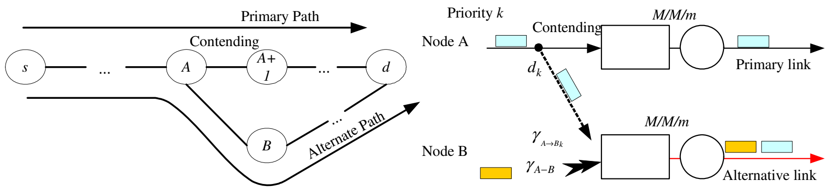
The rest of the paper is organized as follows. The adaptive parameter and optimization rule are defined in Section 2, where ***APDR*** is also proposed. Section 3 illustrates ***APDR***'s results and comparisons with its counterparts through numerical simulations. Finally, this paper is summarized in Section 4.

2 ***APDR***: The Adaptive Parameter Deflection Routing Algorithm

OBS network can be represented as a connected graph $G(N, E)$, where N represents its nodes and E represents its links. $\{D_{ij}\}$ denotes the distance matrix of $G(N, E)$. Assumed that the wavelength each link can support is m, and the $k+1$-*th* priority of burst has precedence over the k-*th* one.

[*] Corresponding author.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1074 – 1079, 2006.



(a) The deflection routing in the link *A-B*

(b) The analysis model of the link *A-B*

Fig. 1. The analytical model of the *APDR* scheme

As a node *A* illustrated by *Fig.1a*, the contention probability $L_{A,\ A+1}$ of the primary link *(A, A+1)* will worsen with the increase of traffic load. The contending bursts can be deflected to other alternate link (e.g., link *(A, B)*) if it has available resource. Obviously, this can reduce $L_{A,\ A+1}$, and increase the utilization rate of link *(A, B)*. However, when the traffic load exceeds a certain threshold, this positive effect will fade away because the premise of deflection exists no more[4]. Here, an adaptive parameter is introduced to control the deflection, which is the deflection probability $d_{A,A+1_k}$ for the contending burst of priority *k* in the link *(A, A+1)*. Naturally, if the priority of contending burst is higher, and its load is heavy, then the deflections should be restricted more strictly because of the resource preempting of high priority deflected burst over low priority normal bursts. Based on the requirements, the parameter can be simply defined by the following expression.

$$d_{A,A+1_k} = (1 - r_{A,A+1_k}^{\theta_k / k}) \cdot L_{A,A+1_k} \qquad \forall k \in [1, N] \tag{1}$$

where $r_{A,A+1_k}$ denotes the ratio of the individual load of *k-th* priority burst to the overall one, and $\sum_{k=1}^{N} r_{A,A+1_k} = 1 \cdot L_{A,A+1_k}$ denotes *BLP* of the *k-th* priority burst.

From *(1)*, we can observe that the parameter for the lower priority burst is higher. However if its parameter is too high, it is possible that the deflection operation can obtain only a little insignificant *BLP* improvement relative to the resources consumed by the low priority deflected burst. Therefore, it is necessary to make a tradeoff between the priority and the parameter. Here, the tradeoff is obtained through a damper factor θ_k of the *k-th* priority burst, which can adjust the sensitivity of the parameter to the burst priority. Of course, the adjustment should satisfy the constraint.

$$\sum_{k=1}^{N} (1 - r_{A,A+1_k}^{\theta_k / k}) = 1 \tag{2}$$

As illustrated *Fig.1a*, the node *A* deflects the contending burst in the probability $d_{A,A+1_k}$ to an optimal deflection path from the contending node *A* to destination node *D* via the node *B*. Here to easily describe the deflection path, a Boolean variable x_{A,B_k} is defined as follows.

$$x_{A,B_k} = \begin{cases} \mathbf{1} & \textit{if } Link(A,B) \in Alternat\_Path(A,D) \\ \mathbf{0} & \textit{otherwise} \end{cases} \tag{3}$$

Then a constraint for the deflection path can be deduced as follows.

$$\sum_{j\in \mathrm{N}} x_{A,j_k} - \sum_{i\in \mathrm{N}} x_{i,A_k} = \begin{cases} 1 & \textit{if } \mathrm{A}=\mathrm{s} \\ -1 & \textit{if } \mathrm{A}=\mathrm{d} \\ 0 & \textit{otherwise} \end{cases} \qquad \forall A,s,d \in \mathrm{N}, \textit{and } \forall k \in [1,N] \tag{4}$$

Assumed that the initial loads in *(A, A+1)* and *(A, B)* are $\gamma_{A,A+1}$ and $\gamma_{A,B}$, respectively. If the *k-th* priority bursts is deflected to the node *B*, the deflected load will add to the link *(A, B)*, which can be expressed as follows

$$\gamma_{A\to B_k} = (r_k \gamma_{A,A+1}) \cdot L_{A,A+1_k} \cdot d_{A,A+1_k} \qquad \forall k \in [1,N] \tag{5}$$

Known from [7]-[10], if the offset-time difference between different burst priorities is enough, the overall performance of OBS network can keep steady regardless of the number of priorities. Therefore from *Fig. 1(b)*, we can get the *BLP* of priority *k* in the link *(A, B)* after its deflection.

$$L_{A,B_k} = \frac{B(\sum_{i=1}^{k}(\gamma_{A\to B_i} + \gamma_{A,B}), m) - \sum_{i=1}^{k-1} C_{A,B_i} L_{A,B_i}}{C_{A,B_k}} \tag{6}$$

where *B(·)* denotes *Erlang B* formula, and C_{A,B_k} denotes the ratio of the individual load of *k-th* priority burst to the overall one in the link *(A, B)* after deflection, written as follows,

$$C_{A,B_k} = \frac{\gamma_{A\to B_k} + \gamma_{A,B} \cdot r_{A,B_k}}{\gamma_{A,B} + \sum_{i=1}^{N} x_{A,B_i} \gamma_{A\to B_i}} \tag{7}$$

Similarly, the overall *BLP* $L_{A,B}$ in the link *(A, B)* after deflection can expressed as follows,

$$L_{A,B} = B(\gamma_{A,B} + \gamma_{A\to B_k}, M) \tag{8}$$

In deflection routing, it is possible that *DB* (data burst) abnormally arrives at the intermediate nodes prior to its corresponding *BHP* (burst head packet). So these *DBs* must be dropped due to the ***offset-time deficit*** resulting in the previous deflection efforts to be fruitless. For the problem, [3]-[4] proposed one solution, i.e., *FDL* (Fiber Delay Line) buffering. However since it is unknown whether and where a burst to have conflict with others, the location and quantity of *FDL* to be configured cannot be decided. Obviously, it is too difficult to resolve it by *FDL* buffering[3]-[4]. Here, the paper tries to resolve it by the nonlinear integer programming to search an optimal deflection path. Naturally, it can be regarded as the following constraint of the optimal path.

$$\sum_{i,j} x_{i,j_k}(D_{i,j}+t_p)\le\delta_k \qquad \forall i,j\in \mathrm{N}, \forall k\in[1,N] \tag{9}$$

where t_p and δ_k denote the *BHP* process time and the initial offset-time, respectively. After δ_k and $D_{i,j}$ normalized by t_p, $D_{i,j}+t_p$ and δ_k can simplify to $D'_{i,j}$ and δ'_k, respectively. Then *(9)* can be reduced as follows,

$$\sum_{i,j} x_{i,j_k} D'_{i,j} \le \delta'_k \qquad \forall i,j\in \mathrm{N} \tag{10}$$

In terms of *BLP* and the e2e delay involved with the deflection interference, we can design the objective function to formulate the optimal deflection path by the similar method of [5], which can be stated as follows,

$$Min \quad \sum_{i,j}[x_{i,j_k}\gamma_{i\to j_k}(D'_{i,j}+L_{i,j_k}{}^{\varepsilon_k})+x_{i,j_k}\gamma_{i,j}(D'_{i,j}+L_{i,j}{}^{\varepsilon})] \tag{11}$$

where ε_k and ε denote the individual and overall burst-loss cost factor, respectively, which can adjust the contribution of the burst-loss to the optimization of deflection path. Under the constraints *(2)*, *(4), (9)-(10)*, we can obtain optimum solution { x_{i,j_k} } to the object function *(11)*, which means that we find optimal deflection path.

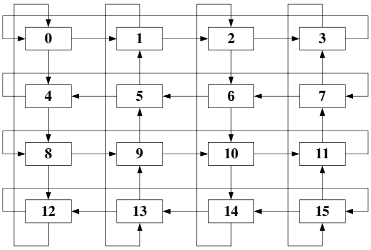
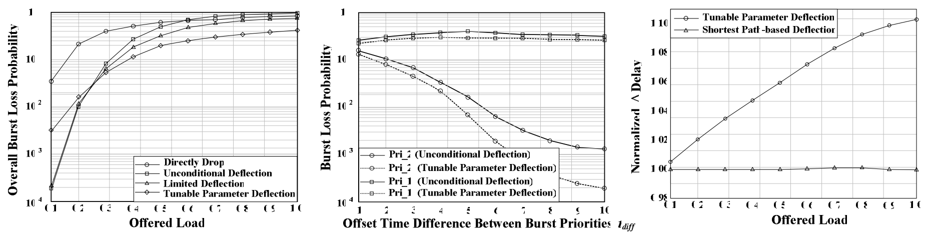


Fig. 2. 4×4 Manhattan street-based simulation network

3 Performance Study and Simulation Numerical Results

The following simulation will evaluate the performance of the proposed scheme in terms of *BLP* and the *e2e delay* by the comparisons with *Directly Drop*, *Unconditional Deflection*[4] and *Limited Deflection*[8]. Assumed that the simulation network is a 4×4 Manhattan street model illustrated by *Fig.2*, in which each link can support 4 wavelengths, its distance is one unit, and its data rate is 10Gbps. For simplicity of analysis, the burst is Poisson arrival, and its length *L* is fixed to 1Mbit. It can support 2 priorities, and the load ratio is assumed to be 0.5.

For different decision of deflection condition, *Fig.3(a)* illustrates their effects in terms of the overall *BLP* under different traffic load. Obviously, when traffic load is not heavy, i.e., $\rho<0.6$, the behavior of *Directly Drop* is the worst. Meanwhile the traffic load is much heavier, *APDR* can adaptively adjust the deflection probability



(a)The traffic load ρ versus the overall *BLP* $(\theta=\varepsilon=\varepsilon_t=1)$

(b)The individual *BLP* versus t_{diff} $(\theta=\varepsilon=\varepsilon_t=1, \rho=0.6)$

(c) The traffic load ρ versus the average e2e delay $(\theta=\varepsilon=\varepsilon_t=1)$

Fig. 3. The performance comparisons of *APDR* with other schemes

according to the traffic load while *Limited Deflection* is insensitive to the variation of traffic load. Naturally, *APDR* behaves better than *Limited Deflection*. Compaered with *Directly Drop*, *APDR* can obtain the maximum gain *47%* in terms of the overall *BLP*.

Next, let us further compare the QoS guarantee capacity between *Unconditional Deflection* and *APDR* under different offset-time difference t_{diff}. Assumed that their QoS schemes are based on the offset-time. As illustrated *Fig.3(b)*, when $BLP=1.1\times10^{-3}$, *APDR* can support the differentiated service if t_{diff} is about *6L*. But for unconditional deflection, its t_{diff} is about *8L*. For other *BLP* case, there is the same trend, i.e., the t_{diff} for unconditional deflection is more than t_{diff} for *APDR*.

Finally comparing with the shortest-path based deflection, we will evaluate the end-end delay of *APDR*. Here, we concern the end-end delay suffered only by the burst successfully arriving at the destination. Certainly, the delay of the shortest-path-based deflection is less than that of *APDR*. As illustrated *Fig.3(c)*, the delay of *APDR* increases along with the traffic load. At the worst case, the delay of *APDR* is higher than that of the shortest-path based deflection about *10%*. It shows that *APDR* behaves very well in terms of its end-end delay. This benefit roots in the item $(\gamma_{i\to j_k}\cdot D'_{i,j})$ in the expression *(11),* which can adjust the relationship between the deflection path length and the input traffic load, that is, if traffic load is heavier, then the deflected burst should choose much shorter deflection path.

4 Conclusion

This paper proposed an adaptive parameter-based deflection routing algorithm, called as *APDR*, which can adaptively adjust the deflection probability according to the traffic load and the burst priority. Using the method in literature [5], this paper designed an object function based on the deflection probability. Under the three constraints (2), (4) and (10), we can find an optimum deflection path derived from the linear programming solutions.

The simulation results show that *APDR* outperforms *directly drop*, *unconditional deflection* and *the limited deflection* in the improvement of the overall *BLP* and the guarantee of differentiated service. In addition, *APDR* can efficiently circumvent the offset-time deficit, and requires less t_{diff} to support QoS. This is very helpful to reduce the end-end delay of *APDR*, which cannot exceed that of the shortest-path-based deflection about 10%.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No.90304004, Hi-Tech Research and Development Program of China (863) under Grant No. 2005AA122310, the Program for New Century Excellent Talents in University (NCET) of the Ministry of Education of China, the Project of the Education Council of Chongqing, and the Projects of the Science and Technology Council of Chongqing (2005BB2062, 2005AC2089).

References

[1] X. Y. Yang, M. R. Dang, Y. J. Mao, and L. M. Li, "A New Burst Assembly Technique for Supporting QoS in Optical Burst Switching Networks", Chinese Optics Letters, 1(5): 266-268, May 2003.

[2] M. Yoo, C. Qiao, S. Dixit, "A comparative study of contention resolution policies in optical burst switched WDM networks", SPIE vol. 4213, pp. 124-135.

[3] Hsu, T. Liu, N. Huang, "Performance analysis of deflection routing in optical burst-switched networks", Proc. of IEEE Infocom' 02 (New York, USA, June 2002), 1: 66-73

[4] X. Wang, H. Morikawa, T. Aoyama, "Burst optical deflection routing protocol for wavelength routing WDM networks", Proc. of SPIE Opticomm'2000, 257-266,Oct. 2000.

[5] S.K. Lee, H.S. Kim, J.S. Song, D. Griffith, "A Study on Deflection Routing in Optical Burst-Switching Networks", Photonic Network Communications, 6(1): 51-59

[6] Y. Chen, H. Wu, D. Xu, C. Qiao, "Performance Analysis of Optical Burst Switched Node with Deflection Routing", Proc. of IEEE ICC, 2: 1355-1359, 2003

[7] M. Yoo, C. Qiao, S. Dixit, "QoS performance in IP over WDM networks", IEEE Journal on Selected Areas in Communications, 18(10): 2062-2071, October 2000.

[8] H. Kim, S. Lee, J. Song, "Optical burst switching with limited deflection routing rules", IEICE Transactions on Communications, E86-B(5): 1550-1554, 2003

[9] K. Dolzer, C. Gauger, J. Späth, S. Bodamer, "Evaluation of reservation mechanisms for optical burst switching", AEÜ Int. J. Electron. Commun., 55(1): 1-8, Jan. 2001.

[10] H. Vu, M. Zukerman, "Blocking probability for priority classes in optical burst switching networks", IEEE Communications Letters. 6(5): 214-216, May 2002

Bandwidth Utilization in Sorted-Priority Schedulers

Tae Joon Kim

Kongju National University, 275 Budae-Dong, Cheonan, Chungnam, 330-240, Korea
tjkim@kongju.ac.kr

Abstract. This paper first introduces bandwidth utilization metric and then analyzes sorted-priority schedulers in the terms of the metric. The results show that the utilization is directly proportional to both the number of delay bound classes and the dependency of delay bound on rate but inversely proportional to packet size.

1 Introduction

Packet scheduling algorithm has been extensively studied in the last decade due to its importance in the provision of Quality of Service (QoS) guarantees. Numerous sorted-priority scheduling algorithms have been developed to emulate the ideal algorithm called General Processor Sharing (GPS) [1]: Weighted Fair Queuing (WFQ) [2] has an ideal latency property with the complexity of O(V). The extreme complexity has been significantly reduced in Self-Clocked Fair Queuing (SCFQ) [3] with sacrificing the latency. The theory of Rate Proportional Schedulers (RPS) [4] was formulated to reduce the complexity without deteriorating the latency, and then applied to various schedulers [5][6]. Now, sorted-priority schedulers become to achieve both the latency of WFQ and the complexity of $O(logV)$.

For each flow i with the desired rate of r_i and the maximum packet size of M_i, its latencies in RPS based and SCFQ schedulers, denoted by Q_i^{RPS} and Q_i^{SCFQ}, respectively, are expressed as follows: $Q_i^{RPS} = M_i/r_i + M/G$[4] and $Q_i^{SCFQ} = M_i/r_i + \sum_{k=1,k\neq i}^{V} M_k/G$[3], where M is the maximum packet size in the scheduler, G is the capacity of outgoing link termed scheduler bandwidth and V is the maximum number of flows that the scheduler can admit. When the latency violates required delay bound, the scheduler should reduce it with even raising the rate reserved for the flow and consequently the bandwidth corresponding to the raised rate will be lost. This loss can not be, unfortunately, evaluated by the three legacy metrics of latency, complexity and unfairness used in previous works [2-6]. In this paper, we first introduce bandwidth utilization metric and then analyze sorted-priority schedulers in the terms of the metric.

2 Utilization Metric

From the latency equation of Q_i^{RPS} we can see that scheduler can improve the latency of each flow as much as it needs with raising its reserved rate. Thus,

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1080–1085, 2006.

bandwidth utilization may be more useful than the latency used as a key metric in previous works [2-6], and, in addition, it enables for us to exactly compute the effective capacity of the scheduler. We define the bandwidth utilization ρ in a scheduler as the ratio of the amount of bandwidth practically used in servicing traffic flows requiring QoS guarantees to the amount of scheduler bandwidth reserved for them.

Now, let us derive a general expression of the bandwidth utilization. For each flow i with the desired rate of r_i, critical rate r_i^{crt} and reservation rate r_i^{res} are introduced: r_i^{crt} is the minimum rate needed to satisfy the delay bound that the flow requires and r_i^{res} is the rate that the scheduler should reserve for the flow in order to simultaneously guarantee both desired rate and required delay bound. r_i^{res} is equal to $max(r_i, r_i^{crt})$. Bandwidth loss due to the excess reservation rate of $(r_i^{res} - r_i)$ is termed reservation loss. Let us define the desired rate of each flow as a random variable R distributed within $[a, b)$ and $r^{req} \equiv E[R]$. Then

$$\rho = \frac{1}{G}\sum_{i=1}^{V} r_i = \frac{r^{req}}{G}, \; where \; V = \{k | \sum_{i=1}^{k} r_i^{res} \leq G \;\; \& \;\; \sum_{i=1}^{k+1} r_i^{res} > G\} \tag{1}$$

3 Utilization Analysis

Let us first consider a scheduler supporting only single delay bound of B second. For simplicity, it is assumed that flows use all the same maximum packet size of M. Then flows have all the same critical rate r^{crt} written as $r^{crt} = GM/(GB - M)$ from the equation of Q_i^{RPS}. The reservation rate of each flow also becomes a random variable of $max(R, r^{crt})$. Since V is recast as $\lfloor G/r^{res} \rfloor$, where $r^{res} \equiv E[max(R, r^{crt})]$, $\rho = (r^{req}/G)\lfloor G/r^{res} \rfloor$.

Next, consider a scheduler in which L delay bounds of B_1, B_2, .., and B_L seconds are supported, where $B_1 > B_2 > .. > B_L$. The set of flows requiring B_n is termed class n traffic and so the designated class number of each flow requiring B_n becomes n. The scheduler can be decomposed into L sub-schedulers as shown in Fig. 1, in which each Sub-Scheduler n (SS n) services only traffic belonging to the corresponding class, i.e., class n traffic. Every flow arrived at the scheduler goes to the sub-scheduler servicing its designated class traffic. Thus the bandwidth utilization can be computed as the weighted sum of the utilizations in L sub-schedulers by the amounts of scheduler bandwidth allocated to them. The utilization in each sub-scheduler can be obtained by the same way as that in the scheduler with single delay bound if both the amount of scheduler bandwidth allocated to the sub-scheduler and the desired rate distribution of flow being arrived at the sub-scheduler are known.

We first develop a framework to obtain the desired rate distribution of flow being arrived at each sub-scheduler, and then analyze the bandwidth utilization.

3.1 Desired Rate Distribution

Every flow being arrived at a scheduler can be characterized by two variables of desired rate and designated class number. We define the designated class number of an arriving flow as a random variable on the sample space $S(L)$, where $S(L)$ is the set of all integers within $[1, L]$. Then, the arriving flow can be represented as a two-dimensional random vector whose components R and N have the joint-probability $f_{R,N}$. In general, f_R is known from the traffic load condition. f_N is, however, not easy to formalize because the designated class of each arriving flow may be relying on various factors such as its desired rate, the number of nodes along its end-to-end path, and the end-to-end delay bound of the service application to which it belongs. For simplicity, only the relation between designated class and desired rate is considered in this paper.

We first formulate a methodology to obtain $f_{N|R}$ meaning the dependency of designated class number on desired rate, in which a rate transformer and filter array shown in Fig. 1 are used. The desired rate of each arriving flow is transformed to an intermediate rate such that the designated class number of the flow becomes a function proportional to the intermediate rate. Then the designated class number n can be expressed as an increasing stepwise function of intermediate rate $\hat{r}$, i.e., $n = ku(\hat{r} - S_k)u(E_k - \hat{r})$, where $k \in S(L)$, $u(\hat{r})$ is a unit step function, $S_1 = a$ and $E_L = b$. Since the range of intermediate rate to be mapped to each class n is $[S_n, E_n)$, a class n filtering function $F^n(\hat{r})$ that extracts the class n traffic can be defined as $F^n(\hat{r}) \equiv u(\hat{r} - S_n)u(E_n - \hat{r})$. We define the intermediate rate of the desired rate r as a random variable $\hat{R}_r$ on the same sample space as that of the random variable R. Then the flow with the desired rate of r can be represented as $f_{\hat{R}_r}(\hat{r})$ in the terms of the intermediate rate $\hat{r}$ as shown in Fig. 1, where $f_{\hat{R}_r}(\hat{r})$ is the probability density function of the random variable $\hat{R}_r$. Note that Fig.1 illustrates an example of how the designated class

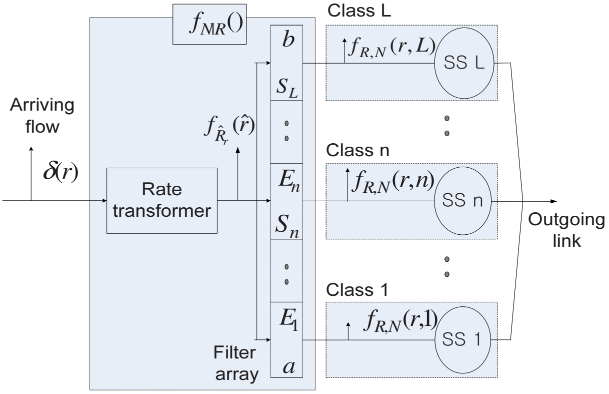


Fig. 1. Internal model of scheduler with L delay bounds

of an arriving flow with the desired rate of r, represented by $\delta(r)$, is determined. Therefore $f_{N|R}$ can be obtained as $f_{N|R}(n|r) = \int_a^b F^n(\hat{r}) f_{\hat{R}}(\hat{r}) d\hat{r}$.

Finding $f_{\hat{R}_r}$ is beyond this paper. Instead we introduce an intermediate rate with the following $f_{\hat{R}_r}(\hat{r})$, termed ϕ intermediate rate, to analyze schedulers with L delay bounds: $f_{\hat{R}_r}(\hat{r}) = \frac{1}{\phi(b-a)}$ for $r - \phi(r-a) \leq \hat{r} \leq r + \phi(b-r)$ and $f_{\hat{R}_r}(\hat{r}) = 0$ otherwise. ϕ is an independency factor indicating the degree that designated class gets free of desired rate: If $\phi = 0$, $f_{\hat{R}_r}(\hat{r})$ is equal to $\delta(r)$ which means that the designated class number of an arriving flow with the desired rate of r becomes a deterministic one proportional to the desired rate r itself. As ϕ increases, the number becomes a random one distributed within more various class numbers because the intermediate rate of the desired rate is more widely distributed. If $\phi = 1$, then it becomes a random number distributed within because the intermediate rate is uniformly distributed within $[a, b)$. For each arriving flow with a desired rate, as its designated class number distributes more widely within $S(L)$, the desired rate of flow being arrived at each sub-scheduler also distributes more widely within $[a, b)$, and then the scheduler will suffer from higher excess reservation rate. Thus the amount of reservation loss in each sub-scheduler will increase with raising ϕ.

Under the ϕ intermediate rate $f_{N|R}(n|r)$ can be developed as $f_{N|R}(n|r) = \int_a^b F^n(\hat{r}) d\hat{r} = \int_a^b u(\hat{r} - B_n) u(\hat{r} - E_n) f_{\hat{R}_r} d\hat{r}$. The desired rate distribution $f_{R|N}(r|n)$ of flow being arrived at each sub-scheduler n is finally obtained as follows: $f_{R|N}(r|n) = f_{N|R}(n|r) f_R(r) / \int_a^b f_{N|R}(n|r) f_R(r) dr$.

3.2 Bandwidth Utilization

Before analyzing the utilization, let us consider how to distribute the scheduler bandwidth to L sub-schedulers. Two policies are possible: explicit allocation in which the amount of bandwidth allocated to each sub-scheduler is previously determined, and implicit allocation in which the amount is implicitly determined by the property of traffic load.

Let us investigate the expected desired and reservation rates, denoted by r_n^{req} and r_n^{res}, respectively, of flow being arrived at each sub-scheduler n. Since r_n^{req} is the conditional expectation of the random variable R given $N = n$, $r_n^{req} = E[R|N = n]$. Because of the same maximum packet size of M, flows within each class n also have all the same critical rate r_n^{crt} written as $r_n^{crt} = GM/(GB_n - M)$ from the equation of Q_i^{RPS}. Then the reservation rate also becomes a random variable of $max(R, r_n^{crt})$. Thus $r_n^{res} = E[max(R, r_n^{crt})|N = n]$.

Now, let us obtain the bandwidth utilization ρ_E in a scheduler with the explicit allocation. Let G_n^E denote the amount of scheduler bandwidth allocated to each sub-scheduler n. Then the sub-scheduler can use the bandwidth of G_n^E regardless of other sub-schedulers and so it can be regarded as a scheduler with single delay bound. Thus from (1) ρ_E can be written as $\rho_E = \frac{1}{G} \sum_{n=1}^{L} G_n^E \rho_E^n$, where $\rho_E^n \equiv (r_n^{req}/G_n^E) \lfloor G_n^E / r_n^{res} \rfloor$.

Meanwhile, sub-schedulers under the implicit allocation share the scheduler bandwidth without any regulation until it will be exhausted. In other words, the scheduler bandwidth becomes a kind of common resource for them. In a consequence, the expected reservation rate r_I^{res} of an arriving flow at the scheduler becomes equal to the sum of the expected reservation rates of flows going to L sub-schedulers. Since the expected reservation rate of flow going to each sub-scheduler n means the weighted expected reservation rate of flow being arrived at the sub-scheduler n by $f_N(n)$, $r_I^{res} = \sum_{k=1}^{L} r_k^{res} f_N(k)$. Thus the bandwidth utilization ρ_I under the implicit allocation can be written as $\rho_I = (r_{req}/G)\lfloor G/r_I^{res} \rfloor$.

4 Numerical Evaluation

Uniformly distributed desired rate within [2, 2048) $Kbps$ and scheduler bandwidth of $10Gbps$ are considered. The bandwidth utilizations of RPS based and SCFQ schedulers for two typical delay bounds under single delay bound are compared in Fig.2. It shows that the utilization decreases with increasing the packet size. This is because longer packet size brings about higher critical rate resulting in larger reservation loss. We observe that the RPS based scheduler has better utilization by up to 50 % than that of the SCFQ one.

Next, RPS based scheduler with L delay bounds is evaluated under the following additional considerations: the delay bound class of each arriving flow is determined both the ϕ intermediate rate and the filter array such that the rate range of each filter n, i.e., $[S_n, E_n)$ is set as $[(r_{n-1}^{crt} + r_n^{crt})/2, (r_n^{crt} + r_{n+1}^{crt})/2]Kbps$, where $r_n^{crt} = 2 + 2048n/(L+1)$, $r_0^{crt} = 2a - r_1^{crt}$ and $r_{L+1}^{crt} = 2b - r_L^{crt}$. Each class has equal bandwidth share under the explicit allocation, i.e., $G_n^E = G/L$ for all $n \in S(L)$. The results are plotted in Fig. 3. We can observe that the utilization becomes better with increasing L or decreasing ϕ, which is due to reducing the reservation loss, and the explicit allocation policy yields some better utilization.

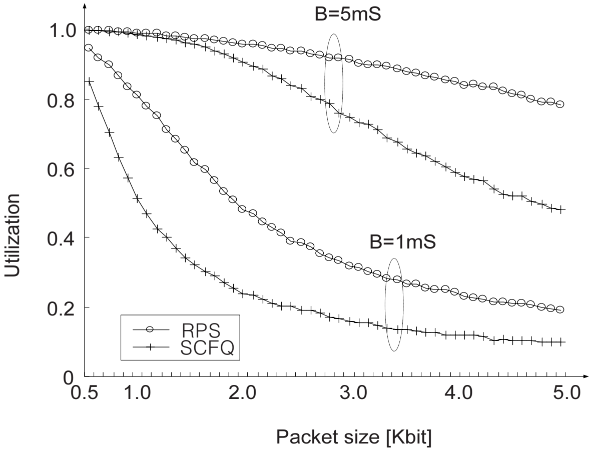


Fig. 2. Bandwidth utilization for single delay bound

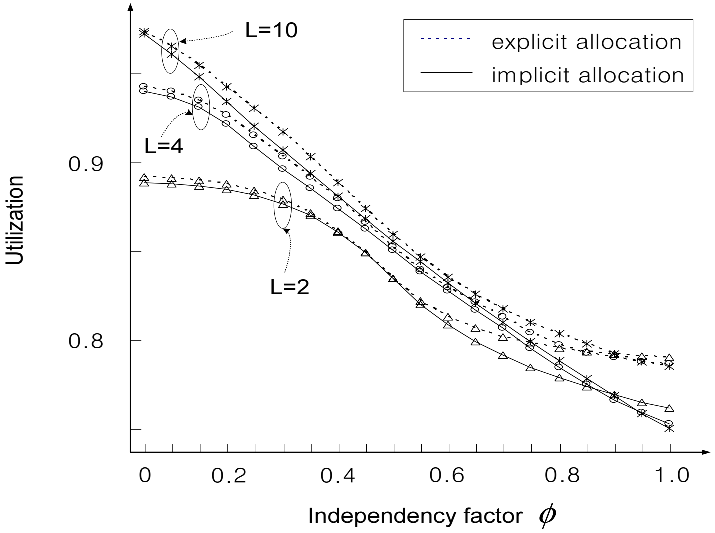


Fig. 3. Bandwidth utilization for L delay bounds

5 Conclusions

In this paper sorted-priority schedulers were analyzed in the terms of bandwidth utilization. A methodology to obtain the delay bound class of an arriving flow from the dependency of delay bound on desired rate was formulated and then used in evaluating the performance of scheduler with multiple delay bounds. The numerical evaluation showed that the bandwidth utilization is directly proportional to both the number of delay bound classes and the dependency of delay bound on rate but inversely proportional to packet size. In particular, schedulers with the latency property of WFQ had much better bandwidth utilization by up to 50 % than that in the SCFQ one.

References

[1] Parekh, A.K.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks. PhD dissertation MIT (1992)

[2] Demers, A., Keshav, S., Shenker, S.: Design and analysis of a fair queuing algorithm. Proc . ACM SIGCOMM (1989) 1-12

[3] Golestani, S.J.: A Self-Clocked Fair Queuing Scheme for Broadband Applications. Proc. IEEE INFOCOM (1994) 636-646.

[4] Stiliadis , D., Varma, A.: Rate Proportional Schedulers: A Design Methodology for Fair Queueing Algorithms. IEEE/ACM Trans. Net. **6** (1998) 164-174

[5] Stiliadis , D., Varma, A.: Efficient Fair Queuing Algorithms for Packet-Switched Networks. IEEE/ACM Trans. Net. **6** (1998) 175-185

[6] Kwak, D., Ko, N., Park, H.: Medium Starting Potential Fair Queueing for High-Speed Networks. IEICE Trans. Commu. **E87-B** (2004) 188-198

A Multicast Approach for UMTS: A Performance Study

Antonios Alexiou, Dimitrios Antonellis, and Christos Bouras

Research Academic Computer Technology Institute, N. Kazantzaki str,
26500 Patras, Greece and
Computer Engineering and Informatics Department,
University of Patras, 26500 Patras, Greece
alexiua@cti.gr, antonel@ceid.upatras.gr, bouras@cti.gr

Abstract. In this paper, a multicast scheme for UMTS which only requires insignificant modifications in the current UMTS network infrastructure is analyzed. We analytically present the multicast routing mechanism behind our scheme as well as the multicast group management functionality of it. Furthermore, we present an evaluation of our scheme in terms of its performance. The critical parameters for the evaluation of the scheme are the number of multicast users within the multicast group, the amount of data sent to the multicast users, the density of the multicast users within the cells and the type of transport channel used for the transmission of the multicast data over the air.

1 Introduction

UMTS constitutes the third generation of cellular wireless networks which aims to provide high-speed data access along with real time voice calls. Wireless data is one of the major boosters of wireless communications and one of the main motivations of the next generation standards [7]. The multicast transmission of real time multimedia data is an important component of many current and future emerging Internet applications, such as videoconference, distance learning and video distribution. It offers efficient multidestination delivery, since data is transmitted in an optimal manner with minimal packet duplication [8].

Compared with multicast routing in the Internet, mobile networks such as UMTS pose a very different set of challenges for multicast. First, multicast receivers are nonstationary and consequently, may change their point of attachment to the network at any given time. Second, mobile networks are generally based on a well-defined tree topology, with the nonstationary multicast receivers being located at the leaves of the network tree. It is therefore not appropriate to apply conventional IP multicast routing mechanisms in UMTS.

Several multicast mechanisms for UMTS have been proposed in the literature. In [1], the authors discuss the use of commonly deployed IP multicast protocols in UMTS networks. However, in [2] the authors do not adopt the use of IP multicast protocols for multicast routing in UMTS and present an alternative solution. The scheme presented in [2] can be implemented within the existing network nodes with only trivial changes to the standard location update and packet-forwarding

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1086 – 1091, 2006.

procedures. Furthermore in [3], a multicast mechanism for circuit-switched GSM and UMTS networks is outlined while in [4] an end to end multicast mechanism for software upgrades in UMTS is analyzed. Additionally, the 3rd Generation Partnership Project (3GPP) is currently standardizing the Multimedia Broadcast/Multicast Service (MBMS) [5], [9].

In this paper, we analytically present a multicast scheme for UMTS with the routing mechanism behind the scheme. Additionally, the multicast group management functionality of our mechanism and the performance of the scheme are analyzed. The critical parameters for the evaluation of the scheme are the number of multicast users within the multicast group, the amount of data sent to the multicast users, the density of the multicast users within the cells and the type of transport channel used for the transmission of the multicast data over the air.

2 A Multicast Approach for UMTS

In this section we present an overview of a multicast scheme for UMTS. More specifically, the way that the multicast packets are delivered to a group of mobile users is presented in detail. Additionally, we analyze the packet forwarding and routing mechanism behind the multicast scheme as well as the multicast group management functionality of the scheme.

Fig. 1 shows a subset of a UMTS network consisting of eleven multicast users located in six cells. The BM-SC acts as the interface towards external sources of traffic [5]. In the presented analysis, we assume that a data stream coming from an external PDN through BM-SC, must be delivered to these UEs as illustrated in Fig. 1. For the efficient packet forwarding mechanism, every node of the network (except the UEs) maintains a routing list. In these lists of each node, we record the nodes of the next level that the messages for every multicast group should be forwarded.

With multicast, the packets are finally forwarded to those Node Bs that serve multicast users. Therefore, in Fig. 1, the Nodes B2, B3, B5, B7, B8, B9 will receive the multicast packets issued by the BM-SC. We briefly summarize the steps occurred for the delivery of the multicast packets. Before the transmission of the multicast data, the routing lists of the nodes must be filled with useful information. This procedure can be initialized either from the UEs or from the BM-SC (ex. software upgrades). In the former case, consider a UE that decides to become a member of a multicast service. Thus, it sends an appropriate message to the BM-SC requesting this service. Then, every node located in the path between this UE and the BM-SC, when it receives the message from the UE, updates its routing list and forwards the message to the next node. In the second case, the BM-SC initializes this procedure and since it does not have any information regarding the location of the multicast members, a paging procedure at RA and URA level is necessary for the updating of the routing lists.

Consider that the BM-SC receives a multicast packet and forwards it to the GGSN that has registered to receive the multicast traffic. Then, the GGSN receives the multicast packet and by querying its routing list, it determines which downstream SGSCs have multicast users residing in their respective service areas. In Fig. 1, the GGSN duplicates the packet and forwards it to the SGSN1 and the SGSN2. After both destination SGSNs have received the multicast packet and having queried their routing list,

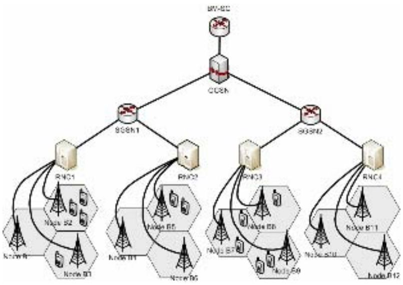


Fig. 1. Packet delivery in UMTS

they determine which RNCs must receive the multicast packet. The destination RNCs receive the multicast packet and send it to the Node Bs that have established the appropriate radio bearers for the multicast application. In Fig. 1, these are Node B2, B3, B5, B7, B8, B9. The transmission of the packets over Uu interface, may be performed on dedicated (DCH) or shared transport channels (ex. High Speed Downlink Shared Channel – HS-DSCH) [7].

3 Evaluation of the Multicast Scheme

In this section we present an evaluation, in terms of the telecommunication costs, of the multicast scheme. In particular, we consider a subset of a UMTS network consisting of a single GGSN and N_{SGSN} SGSN nodes connected to the GGSN. Furthermore, each SGSN manages a number of N_{ra} RAs. Each RA consists of a number of N_{rnc} RNC nodes, while each RNC node manages a number of N_{ura} URAs. Finally, each URA consists of $N_{node\mathrm{b}}$ cells. The total number of RNCs and cells are:

$$N_{RNC} = N_{SGSN} \cdot N_{ra} \cdot N_{rnc} \tag{1}$$

$$N_{NODEB} = N_{SGSN} \cdot N_{ra} \cdot N_{rnc} \cdot N_{ura} \cdot N_{nodeb} \tag{2}$$

The total transmission cost for packet deliveries is considered as the performance metric. We make a further distinction between processing costs at nodes and transmission costs on links. Similar to [6], there is a cost associated with each link and each node of the network for the packet deliveries. We apply the following notations:

D_{gs}	Transmission cost of packet delivery between GGSN and SGSG
D_{sr}	Transmission cost of packet delivery between SGSN and RNC
D_{rb}	Transmission cost of packet delivery between RNC and Node B
D_{DCH}	Transmission cost of packet delivery over the air with DCHs
$D_{HS\text{-}DSCH}$	Transmission cost of packet delivery over the air with HS-DSCH
p_g	Processing cost of packet delivery at GGSN
p_s	Processing cost of packet delivery at SGSN
p_r	Processing cost of packet delivery at RNC
p_b	Processing cost of packet delivery at Node B

The total number of the multicast UEs in the network is denoted by N_{UE}. For the cost analysis, we define the total packets per multicast session as N_p. Furthermore, network operators will typically deploy an IP backbone network between the GGSN, SGSN and RNC. Therefore, the links between these nodes will consist of more than one hop. Additionally, the distance between the RNC and Node B consists of a single hop ($l_{rb} = 1$). In the presented analysis we assume that the distance between GGSN and SGSN is l_{gs} hops, while the distance between the SGSN and RNC is l_{sr} hops.

In multicast, the SGSN and the RNC forward a single copy of each multicast packet to those RNCs or Node Bs respectively that serve multicast users. After the correct multicast packet reception at the Node Bs that serve multicast users, the Node Bs transmit the multicast packets to the multicast users via Dedicated or High Speed Shared Transport Channels. The total cost for the multicast scheme is derived from the following equation where n_{SGSN}, n_{RNC} and n_{NODEB} represent the number of SGSNs, RNCs and Node Bs respectively serving multicast users. The parameter X represents the multicast cost for the transmission of the multicast data over the air.

$$M_S = \left[p_g + n_{SGSN}\left(D_{gs} + p_s\right) + n_{RNC}\left(D_{sr} + p_r\right) + n_{NODEB}\left(D_{rb} + p_b\right) + X \right] N_p \tag{3}$$

$$X = \begin{cases} D_{DCH} \cdot N_{UE}, if \quad channel = DCH \\ D_{HS-DSCH} \cdot n_{NODEB}, if \quad channel = HS-DSCH \end{cases} \tag{4}$$

Having analyzed the costs of the multicast scheme, we try to evaluate the cost in function of a number of parameters. The first parameter is the number of the total packets per multicast session (N_p) and the second one is the number of the multicast users (N_{UE}). We assume a more general network configuration than that illustrated in Fig. 1, with $N_{SGSN} = 10$, $N_{ra} = 10$, $N_{rnc} = 5$, $N_{ura} = 5$ and $N_{nodeb} = 5$.

As we can observe from the equations in the previous section, the cost of the scheme depends on a number of other parameters. Thus, we have to estimate the value of these parameters appropriately, taking into consideration the relations between them. The chosen values of the parameters are presented in Table 1.

Table 1. Chosen parameters' values

D_{gs}	D_{sr}	D_{rb}	p_g	p_s	p_r	p_b	D_{DCH}	$D_{HS\text{-}DSCH}$	l_{gs}	l_{sr}	l_{rb}
36	18	6	1	1	1	1	5	3	6	3	1

In our analysis, the values for the transmission costs of the packet delivery over the air with each of the two transport channels are different. More specifically, the transmission cost over the air with DCHs, is greater than the cost of the packet delivery over the air with HS-DSCH. Therefore, we define the following probabilities for the calculation of the number of the UMTS nodes that serve multicast users:

P_{SGSN}: The probability that an SGSN serve multicast users

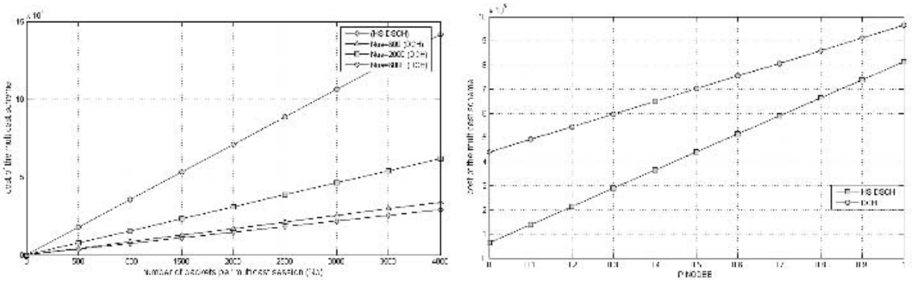
P_{RNC}: The probability that an RNC (served by an SGSN with multicast users), serves multicast users

P_{NODEB}: The probability that a Nobe B (served by an RNC with multicast users), serves multicast users

For the cost analysis, we assume that P_{SGSN}*=0.4*, P_{RNC}*=0.3* and P_{NODEB}*=0.4*. Consequently, the number of the SGSNs, the RNCs and the Node Bs that serve multicast users are $n_{SGSN} = N_{SGSN}\, P_{SGSN} = 4$, $n_{RNC} = N_{RNC}\, P_{SGSN}\, P_{RNC} = 60$ and $n_{NODEB} = N_{NODEB}\, P_{SGSN}\, P_{RNC}\, P_{NODEB} = 600$ respectively.

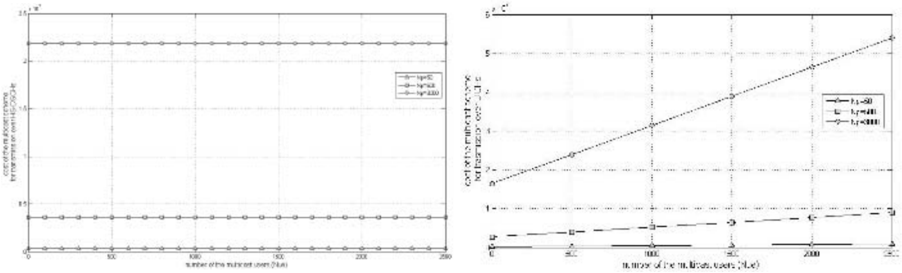
Fig. 2a presents the cost of the multicast scheme in function of the N_p for different transport channels (DCH and HS-DSCH) used for the transmission of the multicast data over the air. The y-axis presents the total cost of the multicast scheme, while the x-axis shows the total packets per multicast session. Regarding the use of DCHs, in Fig. 2a, we have calculated the costs for three different values of the number of multicast users, indicating that the multicast cost increases rapidly when the amount of the multicast data increases. Furthermore, for a given N_p, the multicast cost increases as the members of the multicast group increase. This occurs because the greater the number of multicast users is, the greater the number of DCHs needed for the transmission of the multicast data over the air. Additionally, eqn (3) shows that in case of HS-DSCH, the cost of the multicast scheme depends only on the number of packets per multicast session. This can be shown in Fig. 2a where we can observe that the greater the N_p is, the greater the multicast cost becomes.

Another interesting parameter is the P_{NODEB}. Assuming that N_{UE}*=1500*, N_p*=500*, we can calculate the cost for the multicast scheme, for the transport channels we use. Fig. 2b presents the cost of the multicast scheme in function of P_{NODEB} for different transport channels. It is obvious from Fig. 2b that the cost of the multicast scheme is decreased as P_{NODEB} converges to zero. This means that the greater the number of



a) b)

Fig. 2. Costs of the multicast scheme against *Np* and P_{NODEB} for different transport channels



a) b)

Fig. 3. Costs of the multicast scheme against N_{UE} using different transport channels

multicast users per cell is, the lower the cost of the multicast scheme is. Furthermore as Fig.2b indicates, the use of HS-DSCHs is absolutely preferable than the DCHs.

Furthermore, we try to estimate the cost of the multicast scheme in function of the N_{UE} (Fig. 3). As we observe, three different values of the number of the total packets per multicast session (N_p) have been calculated. Fig. 3a presents the cost of the multicast scheme against N_{UE} in case we use HS-DSCH for the transmission of the multicast data over the air. According to Fig. 3a, the cost of the multicast scheme is independent from the number of multicast users in case of HS-DSCH. The cost of the multicast scheme in this case depends mainly on the number of Node Bs that serve multicast users. Only one per cell HS-DCSH is established and it is capable of supporting a great number of multicast users in the specific cell. Regarding the multicast cost against N_{UE} in case of using DCHs for the transmission of the multicast data over the air, the relation between them is predictable, since the greater the number of the multicast UEs is, the greater the cost becomes (Fig. 3b).

4 Conclusions and Future Work

In this paper, we have presented a multicast scheme for UMTS. We have analyzed the delivery of the multicast packets to a group of mobile users and the performance of such a delivery in terms of the telecommunication cost. Considering a general network configuration, we have presented the cost of a multicast scheme in function of a number of parameters. The step that follows this work is to implement the above presented multicast scheme in NS-2 simulator and confirm the relation of the costs through the experiments.

References

1. Hauge, M., Kure, O.: Multicast in 3G networks: Employment of existing IP multicast protocols in UMTS. in Proc. WoWMoM 2002 96–103
2. Rummler, R., Chung, Y., Aghvami, H.: Modeling and Analysis of an Efficient Multicast Mechanism for UMTS. IEEE Trans. on Vehicular Technology, vol. 54, no. 1(2005) 350-365
3. Lin, Y.: A multicast mechanism for mobile networks. IEEE Communication Letters, vol. 5 (2001) 450–452
4. Rummler, R., Aghvami, H.: End-to-end IP multicast for software upgrades of reconfigurable user terminals within IMT-2000/UMTS networks. IEEE ICC'02, vol. 1 (2002) 502–506
5. 3GPP, TS 23.246, Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description, V6.7.0
6. Ho, J. S., Akyildiz, I. F.: Local anchor scheme for reducing signaling costs in personal communications networks. IEEE/ACM Trans. Networking, vol. 4 (1996) 709–725
7. Holma. H.., Toskala, A.: WCDMA for UMTS: Radio Access for Third Generation Mobile Communications. John Wiley & Sons (2003)
8. Gossain, H., Cordeiro, C. Argawal, D.: Multicast: Wired to Wireless. IEEE Communications Magazine (2002) 116-123
9. 3GPP, TS 22.146, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service, Stage 1 (Release 6)

Echidna: Efficient Clustering of Hierarchical Data for Network Traffic Analysis

Abdun Naser Mahmood, Christopher Leckie, and Parampalli Udaya

Department of Computer Science and Software Engineering,
University of Melbourne, Australia
{abdun, caleckie, udaya}@csse.unimelb.edu.au

Abstract. There is significant interest in the network management community about the need to improve existing techniques for clustering multi-variate network traffic flow records so that we can quickly infer underlying traffic patterns. In this paper we investigate the use of clustering techniques to identify interesting traffic patterns in an efficient manner. We develop a framework to deal with mixed type attributes including numerical, categorical and hierarchical attributes for a one-pass hierarchical clustering algorithm. We demonstrate the improved accuracy and efficiency of our approach in comparison to previous work on clustering network traffic.

1 Introduction

There is a growing need for efficient algorithms to detect important trends and anomalies in network traffic data. In this paper, we present a hierarchical clustering technique for identifying significant traffic flow patterns. In particular, we present a novel way of exploiting the hierarchical structure of traffic attributes, such as IP addresses, in combination with categorical and numerical attributes. This algorithm addresses the scalability problems in previous approaches [5-9] of network traffic analysis as it is a one-pass, fixed memory algorithm.

A key challenge in clustering multi-dimensional network traffic data is the need to deal with various types of attributes: numerical attributes with real values, categorical attributes with unranked nominal values and attributes with hierarchical structure. For example, byte counts are numerical, protocols are categorical and IP addresses have hierarchical structure. We have proposed a hierarchical approach to clustering that exploits the hierarchical structure present in network traffic data. In network traffic a hierarchical relation between two IP addresses can reflect traffic flow to or from a common sub-network. We propose a common framework to incorporate such hierarchical attributes in the distance function of our clustering algorithm.

The second contribution of this paper is the use of a single-pass hierarchical clustering technique to address the problems suffered by existing algorithms in terms of their need to make multiple passes through the dataset. In order to keep the size of the reports small we present a number of summarization techniques over the cluster tree.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1092 – 1098, 2006.

In the next section we briefly summarize existing research on identifying trends in network traffic. In Section 3 we present our clustering and summarization algorithm called Echidna. We demonstrate the effectiveness of our approach using an empirical evaluation in Section 4.

2 Related Work

The problem of identifying network trends has been studied by [5-9]. In [2], the authors address the problem of finding patterns in network traffic by proposing a frequent itemset mining algorithm. Their tool, called AutoFocus [1] identifies significant patterns in traffic flows by using frequent itemset mining. It first creates a report based on unidimensional clusters of network flows and then combines these unidimensional clusters in a lattice to create a traffic report based on multidimensional clusters. AutoFocus requires multiple passes through the network traffic dataset in order to generate *significant* multidimensional clusters. To address this inefficiency, we consider the use of a hierarchical clustering algorithm.

Our approach to finding multidimensional clusters of network data builds on the BIRCH framework [3], which is a clustering algorithm that uses a *Cluster Feature* (CF) to represent a cluster of records in the form of a vector <*n, LS, SS*>, where *n* is the number of records in the cluster, *LS* is the linear sum and *SS* is the square sum of the attributes of the records. Clusters are built using a hierarchical tree called a *Cluster Feature* Tree (CF-Tree) to summarize the input records.

The tree is built in an agglomerative hierarchical manner (see Fig. 1). Each leaf node consists of *l* clusters, where each cluster is represented by its CF record. These CF records can themselves be clustered at the non-leaf nodes. Figure 1 shows a CF-Tree in fixed memory M with branching factor *B* and leaf node capacity *L*. If *P* denotes the size of a node in the tree, then it takes only $O(B^*(1+\log_B M/P))$ comparisons to find the closest leaf node in the tree for a given record [3].

An open issue for using the BIRCH approach to cluster network traffic records is how to cope with numerical, categorical and hierarchical attributes which are used to describe the network traffic. We also require a method for extracting significant clusters from the CF-tree in order to generate a concise and informative report on the

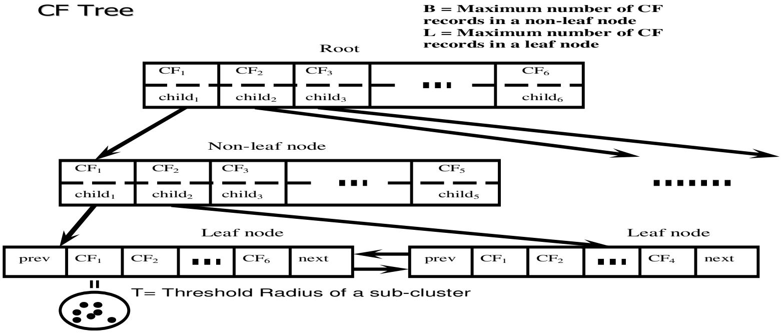


Fig. 1. A Cluster Feature Tree

given network traffic data. In the next section we propose several modifications to the BIRCH clustering approach to address these problems.

3 Our Approach to Clustering Network Traffic: Echidna

The input data is extracted from network traffic as 6-tuple records <*SrcIP, DstIP, Protocol, SrcPort, DstPort, bytes*>, where *SrcIP, DstIP* are *hierarchical* attributes, *bytes* is numerical and the rest are *categorical* attributes. Our algorithm takes each record and iteratively builds a hierarchical tree of clusters called a CF-Tree. We now describe the distance functions used for each attribute type.

Distance Functions: When clustering network traffic records we need to consider three kinds of attributes: *numerical*, *categorical* and *hierarchical.*

a) Numerical Attributes: A *numerical* attribute is represented by a scalar $x[i] \in R$. The centroid $\bar{c}[i]$ of a numerical attribute i in cluster **C** having N points is given by the mean of the N points. We calculate the distance d_n between the centroids of two clusters $\mathbf{C}_1$ and $\mathbf{C}_2$ by using the Euclidean distance metric.
b) Categorical Attributes: In the case of a *categorical* attribute, **x**[i] is a vector$\in Z^c$, where c is the number of possible values that the *categorical* attribute i can take. For a d-dimensional *categorical* attribute vector **X**, let the i^{th} attribute **x**[i] be represented as **x**[i] $=\{a_1, a_2,\ldots,a_c\}$. The centroid $\bar{\mathbf{c}}[i]$of a *categorical* attribute i in cluster **C** having N points is represented by a histogram of the frequencies of the attribute values. The distance between clusters $\mathbf{C}_1$ and $\mathbf{C}_2$ in terms of a single *categorical* attribute is given by the Euclidean distance between the frequency vectors of each attribute.
c) Hierarchical Attributes: A *hierarchical* attribute represents a generalization hierarchy in the form of an L-level tree applied to a domain of values at the leaves of the tree. A non-leaf node in the hierarchy is a generalization of the leaf nodes in the subtree rooted at that node.

In a cluster **C**, the centroid for a *hierarchical* attribute that corresponds to an IP address is represented by an IP prefix $\overline{IP}/p$, which is an aggregate of the IP addresses [10] in that cluster. We calculate the distance between two clusters C_1 and C_2 with centroids $\overline{IP}_1/p_1$ and $\overline{IP}_2/p_2$ as $d_h(C_1, C_2) = 32 - p$ if $p > 8$, or 24 if $p \leq 8$, where $p = CommonPrefix(\overline{IP}_1/p_1, \overline{IP}_2/p_2)$. The definitions of *CommonPrefix* and *IP aggregation* can be found in [10].

Intuitively, d_h corresponds to the hierarchical distance from the leaf level of the tree to the most specific generalization of the two centroids. In the case of IP addresses, this corresponds to the size in logarithm (base 2) of the smallest subset that would be required to contain these two clusters. For example, the distance between 128.0.0.252/32 and 128.0.0.254/31 is 2. The distance between two centroids is the squared sum of the distances of each attribute using the appropriate distance function. Note that each attribute is scaled into the range [0,1] so that no single attribute dominates.

Radius Calculation: In order to control the variance of data records within a cluster, we need some measure of the *radius* of a cluster. The *radius* for *numerical* and *categorical* attributes can be represented in a straightforward manner as the standard deviation of the attribute values of the records in the cluster. In the case of hierarchical attributes, we propose that the *radius* is proportional to the size of the subtree in the *generalization hierarchy* covering the values that appear in the cluster. Consider the case of IP addresses. We keep two variables *minIP* and the *maxIP*, which correspond to the smallest and the largest IP values present in the cluster. Let *C*[i]*.range*=(*minIP*, *maxIP*) denote the range of IP addresses present in attribute *i* of cluster *C*. We can measure this *radius* in terms of the height of the smallest subtree in the *generalization hierarchy* that covers *minIP* and *maxIP*, which can be calculated using *CommonPrefix* as R_h = *(32-CommonPrefix(minIP, maxIP))/32*. The final radius value of the cluster is simply a linear combination of the individual radius values of different attributes types.

Cluster Formation: Following the general approach of BIRCH [3], each cluster C_l is represented by a cluster feature vector that contains sufficient statistics to calculate the centroid $\overline{c}_l$ and *radius* R_l of the cluster. Each data record *X*, corresponding to a 6-tuple traffic flow record, is inserted by comparing *X* to the closest cluster starting from the root along a path *P* to a leaf node. At the leaf node, the data record *X* is inserted into the closest C_l and the *radius* R_l of the updated cluster is calculated. If $R_l > T$, where *T* is a threshold value in the range [0,1], and if the number of CF entries in the node is less than a minimum *m*, then *X* is inserted into the node as a new cluster. If a node has no more space for a new CF entry, then the node is split to create a new node and the path to the root is updated recursively.

Summarization: The clusters at each level represent a generalized set of traffic flows, which can be used to describe the traffic flows in the network. Since there is redundant information between different levels, the summary report should contain only those nodes of any level having significant additional information compared to their descendant levels. We define significant nodes in terms of number of records, *Average Intra-Cluster* distance and *Maximum Intra-cluster* distance measures that intuitively pick those nodes that contain a heterogeneous set of clusters.

An index node is considered significant if one of its descendants is significant. A leaf node is significant if it has the following properties:

a) The number of records in the leaf node *C* is above a certain threshold $T_{r.}$

b) The *Average Intra-cluster* (AI) distance of the leaf node *C* is above a threshold T_{ai}, where the AI distance of cluster *C* with respect to its *l* sub-clusters $C_1,\ldots,C_l$ is

$$AI(C) = \sqrt{\left(2\sum_{i=1}^{l} \sum_{j=2}^{l} (C_i - C_j)^2 / l(l-1) \right)}$$, where C_i and C_j are sub-clusters of *C*

c) The *Maximum Intra-cluster* (MI) distance of the leaf node *C* is greater than or equal to the AI distance. The MI distance is given by MI(C) = $max\{d(C_i, C_j)\}$, i=1,..,l and j=2,..,l

Compression: We require a technique to further compress the number of significant clusters that are included in the final report. We consider a node to be significant if the number of traffic records it contains is greater than T_r. Lemma 1 then gives the upper and lower bound on the number of significant nodes in the tree. The proof of Lemma 1 can be found in [10].

Lemma 1: For a cluster tree of height h with τ traffic records and threshold T_r, the size of the report ρ is bounded by $h\frac{\tau}{T_r} \geq \rho \geq 2\frac{\tau}{T_r}, h \geq 2$.

Compression of cluster report: Let $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2,\ldots,\mathbf{C}_h\}$ be the set of clusters in a path P from the root to a node in level h of the CF-tree. Since a cluster $\mathbf{C}_i$ is represented as a node in the tree, then $\mathbf{C}_i$ consists of a set of l sub-clusters (l CF entries) at the same level i of the tree $\mathbf{C}_i=\{C_{1,i},C_{2,i},..,C_{l,i}\}$. It follows that

a) $\mathbf{C}_i$ is significant if there exists a $\mathbf{C}_j$ in the path P, such that $\mathbf{C}_j$ is significant, where $i < j$, i.e., $\mathbf{C}_i$ is an ancestor of $\mathbf{C}_j$.

b) Let τ_i and τ_j denote the traffic of $\mathbf{C}_i$ and $\mathbf{C}_j$, then $\tau_i \geq \tau_j$, if $i < j$. In other words, the size of cluster $\mathbf{C}_i$ is greater than or equal to the size of cluster $\mathbf{C}_j$.

Let ρ be the compressed report, and $\mathbf{C}_i$ and $\mathbf{C}_j$ are significant clusters. $\mathbf{C}_i$ is included in ρ if τ_i - τ_j > T_r, where T_r is the threshold of records. T_r can be expressed as a proportion of the total traffic size, $T_r = r\tau$, where $r = [0,1]$ and T is the total traffic. In other words, a higher level cluster is only included if it reports some traffic not mentioned by its more specific significant sub-clusters.

Complexity: Since the total number of attributes and their range of values are fixed, we can consider that the cost of distance calculation between a record and a cluster is also constant. In a height-balanced CF-Tree with branching factor B and m nodes, $log_B\, m$ comparisons are required for each record to be inserted into the closest leaf cluster. For N records the insertion time is bounded by $O(N * B(1 + \log_B m))$.

4 Evaluation

Our aim was to test the accuracy and scalability of our hierarchical traffic summarization algorithm. We have compared the accuracy and run-time performance of our algorithm to AutoFocus [2] using 1998 DARPA dataset [4]. Note that the attack/normal labels in this dataset are used for evaluation purposes only, and are not used as part of the cluster formation process.

Detection Accuracy: Our aim is to generate a summary traffic report that identifies important flows in network traffic. In this case, we use the DARPA traces (weeks 3-5) to test whether the reports generated by Echidna or AutoFocus identify specific attacks that appear in the traces. For each file, we identified the number and type of attacks, reported as clusters in the summary reports from Echidna and AutoFocus, and identified the total number of occurrences of these attack types in the traces (see Table 1).

Echidna was able to detect 7 different types of attacks compared to 4 attack types detected by AutoFocus. Moreover, in the case of the ipsweep attack, Echidna detected 3 instances compared to 1 instance detected by AutoFocus. In most cases, the attacks that were detected can be characterized by their influence on the network bandwidth during the time of the attack.

Run-time performance: In order to test the scalability of our algorithm in comparison to AutoFocus, we measured the execution time required by Echidna and AutoFocus for different traffic samples on a time shared dual 2.8GHz Xeon processor machine with 4 GB RAM running SunOS 5.9 (see Fig. 2).

As predicted by the complexity analysis in Section 3, the computational complexity of Echidna is linear with respect to the number of input traffic flow records. Furthermore, Echidna shows a significant reduction in computation time and variance in comparison to AutoFocus.

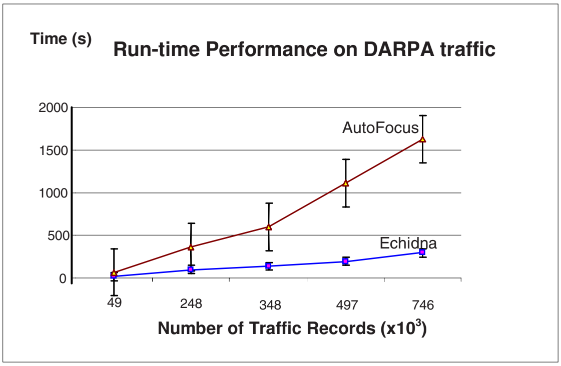


Fig. 2. Comparison of Run-time Performances

Table 1. Detection Accuracy

	Number of Detected Attacks		
Attack	Total	AF	Echidna
ipsweep	5	1	**3**
Neptune	5	4	4
Nmap	2	0	**1**
Pod	7	0	**2**
Satan	4	2	2
Syslog	2	0	**1**
Smurf	7	3	3

5 Conclusion

We have presented a clustering scheme called Echidna for generating summary reports of significant traffic flows in network traces. The key contributions of our scheme are the introduction of a new distance measure for hierarchically-structured attributes, such as IP addresses, and a set of heuristics to summarize and compress reports of significant traffic clusters from a hierarchical clustering algorithm. Using standard benchmark traffic traces, we have demonstrated that our clustering scheme achieves greater accuracy and efficiency in comparison to previous work.

References

1. http://www.caida.org/tools/measurement/autofocus/
2. C. Estan, S. Savage. and G. Varghese. Automatically Inferring Patterns of Resource Consumption in Network Traffic problem. In Proceedings of SIGCOMM 2003
3. T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pages 103-114, 1996
4. http://www.ll.mit.edu/IST/ideval/data/1998/1998_data_index.html
5. A. Medina, K. Salamatian, N. Taft, I. Matta, and C. Diot. A Two-step Statistical Approach for Inferring Network Traffic Demands (Revises Technical Report BUCS-TR-2003-003).

6. A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft. Structural analysis of network traffic flows, In Proceedings of ACM SIGMETRICS, June 2004.
7. A. Lakhina, M. Crovella, and C. Diot. Characterization of Network-Wide Anomalies in Traffic Flows. Technical Report BUCS-2004-020, Boston University, 2004.
8. K. Lan, and J. Heidemann. On the correlation of Internet flow characteristics. Technical Report ISI-TR-574, USC/Information Sciences Institute, July, 2003.
9. K. C. Claffy, G. C. Pluyzos, and H. W. Braun. Applications of Sampling Methodologies to Network Traffic Characterization. In Proceeding of ACM SIGCOMM, 1993.
10. A. Mahmood, C. Leckie, P. Udaya. Echidna: Efficient Clustering of Hierarchical Data for Network Analysis. (http://www.cs.mu.oz.au/~abdun/TR01112005.pdf)

Cross-Layer Performance of a Distributed Real-Time MAC Protocol Supporting Variable Bit Rate Multiclass Services in WPANs

David Tung Chong Wong[1], Jon W. Mark[2], and Kee Chaing Chua[3]

[1] Institute for Infocomm Research, 21 Heng Mui Keng Terrace,
Singapore 119613, Singapore
wongtc@i2r.a-star.edu.sg

[2] Center for Wireless Communications, University of Waterloo,
Waterloo, Ontario, Canada N2L 3G1
jwmark@bbcr.uwaterloo.ca

[3] Electrical and Computer Engineering, National University of Singapore,
10 Kent Ridge Crescent, Singapore 119260, Singapore
chuakc@nus.edu.sg

Abstract. A cross-layer optimization problem to maximize utilization for a distributed real-time medium access control (MAC) protocol supporting variable bit rate (VBR) multiclass services is formulated. A complete sharing (CS) scheme is used as the admission control policy at the connection level to relate the maximum number of devices that can be admitted in the wireless personal area network (WPAN) to the grade of service (GoS) of blocking probability at the connection level, the quality of service (QoS) of packet loss probability at the packet level and the effective data transmission slots efficiency at the MAC layer. With this cross-layer analytical framework, the maximum number of devices that can be admitted into the system to achieve maximum utilization while maintaining prescribed GoS/QoS requirements under different total device mean connection arrival rate can be determined. Numerical results are presented to demonstrate the effectiveness of the proposed cross-layer coupling strategy.

1 Introduction

Cross-layer design, a hot research area [1], aims to optimize performance across different layers of the layered-communications model. By considering cross-layer design, the design system performance can be optimized. In this paper, we consider a basic distributed real-time medium access control (MAC) protocol like the multiband OFDM Alliance (MBOA) MAC [2] without the non-real-time contention MAC protocol for Ultra-Wideband (UWB) systems. A wireless mobile multimedia UWB network has to provide a reasonable user-transparent grade of service (GoS)/quality of service (QoS) for different service classes. To our knowledge, there is no cross-layer performance analysis among utilization, maximum number of devices, blocking probability, packet loss probability, effective data transmission slots efficiency and total device arrival rate in such a distributed real-time MAC protocol to date. The blocking probability is at the connection level in the network layer, while the packet loss probability is at the packet level in the network layer. The effective data transmission slots

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1099 – 1105, 2006.

efficiency is at the MAC sublayer in the link layer. The goal is to find the maximum number of devices that can be supported such that utilization is maximized under reasonable GoS/QoS specifications. The main contribution of this paper is the analytical formulation and evaluation of the cross-layer optimization problem for the support of variable bit rate multiclass services. The cross-layer optimization here is shown to have some system utilization improvement exceeding 100% compared with a simple admission scheme.

2 Distributed Real-Time MAC Protocol

The distributed real-time MAC protocol, e.g., [2], uses a frame format, as shown in Fig. 1. Each frame consists of two parts: a beacon subframe and a data transmission subframe. The former consists of beacon slots for existing devices in the system, which are packed at the beginning of the beacon period subframe, while N_{BC} beacon slots are available for new devices to get into the system through contention using, e.g., the slotted ALOHA protocol. After successful entry into the system, the new devices are packed together at the beginning of the beacon period. The number of contention beacon slots is assumed constant. When devices leave the system, a packing protocol is used to pack the remaining devices' beacon slots together at the beginning of the beacon slots period. Thus the beacon period is not fixed but varies according to the number of devices in the system. The beacon period subframe is assumed to be an integer number of packet slots.

The data transmission period is used to transmit data packets whose data reservations are announced in its device beacon slot. This is called the data reservation protocol (DRP) [2]. Transmission need not be in the same order as the devices in the beacon slots and the DRP packets for each device also need not be transmitted immediately after other DRP packets. The number of transmission data packets for each device is not fixed but can vary. Note that all devices announce their data reservations and each device beacon slot contains information on all other devices [4]. Since the beacon period varies, the available number of data packet slots, C, also varies, depending on the number of devices in the system and the number of contention beacon slots:

$$C = C(n_1, n_2, ..., n_K) = \left(N_{SF} - \left\lceil \left(\sum_{k=1}^{K} n_k + N_{BC} \right) \Big/ N_{BS} \right\rceil \right), \ \sum_{k=1}^{K} n_k \le N_D, \qquad (1)$$

where K is the number of traffic classes, N_{SF} is the number of packet slots in a superframe, n_k is the number of class k devices in the system and N_{BS} is the number of beacon slots equivalent to one packet slot. $\lceil x \rceil$ is the nearest upper integer value of x.

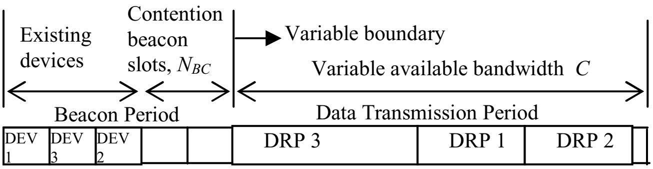


Fig. 1. Frame format of a distributed real-time MAC protocol

3 Analytical Model

At the connection level, the GoS is the blocking probability. In general, this GoS decreases with increase in the maximum number of devices, N_D. At the packet level, the QoS is the packet loss probability. In general, this QoS is zero as long as the maximum total packet transmissions do not exceed the available number of data transmissions in a frame. However, when it starts to increase with the increase in the maximum number of devices, N_D, its initial increase can be at a very sharp rate. This is the limiting constraint in the numerical example in Section 4. At the MAC layer, the QoS is the effective data slots transmission efficiency. It decreases with the increase in the maximum number of devices, N_D. The cross-layer optimization here is shown to have some system utilization improvement exceeding 100% compared with a simple admission criterion in Section 4.

The complete sharing (CS) scheme is used as the resource admission policy. Fig. 2 shows the CS admission Markov chain for 2 classes of devices. λ_k is the class k device mean connection arrival rate, while μ_k is the class k device mean departure rate. $1/\mu_k$ is the class k mean connection holding time. These parameters determine the blocking probability GoS.

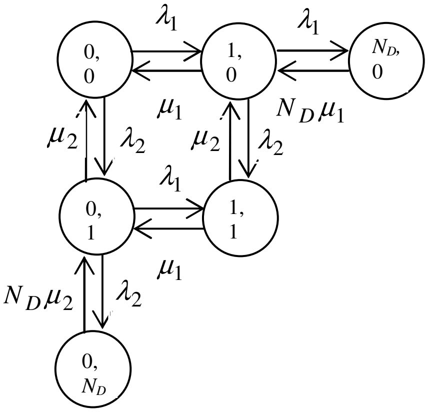


Fig. 2. Complete sharing resource allocation scheme for two classes

The steady state probability of a complete sharing resource allocation scheme for K classes is given by

$$P(n_1, n_2, ..., n_K) = \frac{\frac{(\lambda_1/\mu_1)^{n_1}}{n_1!}\frac{(\lambda_2/\mu_2)^{n_2}}{n_2!}\cdots\frac{(\lambda_K/\mu_K)^{n_K}}{n_K!}}{\sum\limits_{n_1=0}^{N_1}\sum\limits_{n_2=0}^{N_2}\cdots\sum\limits_{n_K=0}^{N_K}\frac{(\lambda_1/\mu_1)^{n_1}}{n_1!}\frac{(\lambda_2/\mu_2)^{n_2}}{n_2!}\cdots\frac{(\lambda_K/\mu_K)^{n_K}}{n_K!}}, \tag{2}$$

where $N_k = N_D - \sum_{i=1}^{k-1} n_i, k = 1,2,..., K$. The N_k's together with the summations determine the state space of the K-dimensional Markov chain. The blocking probability, P_B, is given by

$$P_B = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} ... \sum_{n_K=0}^{N_K} P(n_1, n_2,..., n_K), \text{if } \sum_{k=1}^{K} n_k = N_D . \tag{3}$$

A class k variable bit rate source can be modeled by a continuous-time Markov chain with finite states [3] in Fig. 3.

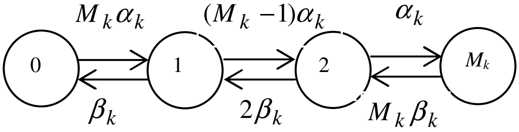


Fig. 3. Continuous-time Markov chain for a single variable bit rate source

Each state represents the discrete level of bit rate generated by a single source. State 1 requires r_k number of packet slots for transmission. The highest state is state M_k. This state requires $M_k r_k$ number of packet slots for transmission. Thus the packet slots variations are in the set of $\{0,r_k,2r_k,3r_k,…,M_k r_k\}$. α_k is the increase rate of one two-state mini-source, while β_k is the decrease rate of one two-state mini-source. The steady-state probability of being in state m_k, denoted by P_{m_k}, is given by

$$P_{m_k} = \binom{M_k}{m_k} (p_k)^{m_k} (1 - p_k)^{M_k - m_k} , \; m_k = 0,1,2,..., M_k , \tag{4}$$

where $p_k = \alpha_k / (\alpha_k + \beta_k)$. The probability that l_k levels of bit rate for class k traffic given that there are n_k sources, denoted by $\Pr[l_k|n_k]$, is given by

$$\Pr[l_k \mid n_k] = \binom{n_k M_k}{l_k} (p_k)^{l_k} (1 - p_k)^{n_k M_k - l_k} , \; l_k = 0,1,2,..., n_k M_k , \tag{5}$$

Assuming real-time traffic with no storing of packets for retransmission, the packet loss probability, P_L, is given by

$$P_L = \frac{\left[\begin{array}{l} \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} ... \sum_{n_K=0}^{N_K} P(n_1, n_2,..., n_K) \sum_{l_1=0}^{n_1 M_1} \sum_{l_2=0}^{n_2 M_2} ... \sum_{l_K=0}^{n_K M_K} \Pr(l_1 \mid n_1) \\ \times \Pr(l_2 \mid n_2) ... \Pr(l_K \mid n_K) \left[\sum_{k=0}^{K} l_k r_k - C(n_1, n_2,..., n_K) \right]^+ \end{array} \right]}{\left[\begin{array}{l} \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} ... \sum_{n_K=0}^{N_K} P(n_1, n_2,..., n_K) \sum_{l_1=0}^{n_1 M_1} \sum_{l_2=0}^{n_2 M_2} ... \sum_{l_K=0}^{n_K M_K} \Pr(l_1 \mid n_1) \\ \times \Pr(l_2 \mid n_2) ... \Pr(l_K \mid n_K) \left[\sum_{k=0}^{K} l_k r_k \right] \end{array} \right]} , \tag{6}$$

where $[x]^+ = \max[0, x]$. The utilization, N_u, is given by

$$N_u = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \ldots \sum_{n_K=0}^{N_K} P(n_1, n_2, \ldots, n_K) \sum_{l_1=0}^{n_1 M_1} \sum_{l_2=0}^{n_2 M_2} \ldots \sum_{l_K}^{n_K M_K} \Pr(l_1 \mid n_1) \times \Pr(l_2 \mid n_2) \ldots \Pr(l_K \mid n_K) U, \tag{7}$$

where

$$U = \begin{cases} \sum_{k=1}^{K} l_k r_k, & \text{if } \sum_{k=1}^{K} l_k r_k \le C(n_1, n_2, \ldots, n_K) \\ C(n_1, n_2, \ldots, n_K), & \text{if } \sum_{k=1}^{K} l_k r_k > C(n_1, n_2, \ldots, n_K) \end{cases}. \tag{8}$$

The probability of n devices in the system, P_n, is given by

$$P_n = \Pr[\sum_{k=1}^{K} n_k = n] = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \ldots \sum_{n_K=0}^{N_K} P(n_1, n_2, \ldots, n_K), \text{if } \sum_{k=1}^{K} n_k = n, n = 0, 1, \ldots, N_D. \tag{9}$$

Thus the effective available capacity, C_e, is given by

$$C_e = \sum_{n=0}^{N_D} P_n C(n_1, n_2, \ldots, n_K), \ \sum_{k=1}^{K} n_k = n. \tag{10}$$

The data slots transmission efficiency, ϕ, is given by

$$\phi = C(n_1, n_2, \ldots, n_K) / N_{SF}, 0 < \phi < 1. \tag{11}$$

Similarly, the effective data slots transmission efficiency, ϕ_e, is given by

$$\phi_e = C_e / N_{SF}, 0 < \phi_e < 1. \tag{12}$$

The system utilization can be maximized by solving the following constraint optimization problem of maximizing N_u subject to the constraints of $P_B \le P_B^*, P_L$, and $\phi_e \ge \phi_e^*$, where the superscript * denotes the requirement values of the corresponding parameters. The results here can be extended for the blocking and loss probabilities of each traffic class.

4 Numerical Results

In this section, we illustrate the system performance by presenting results for a two-class traffic example. The parameter values used in the numerical examples are tabulated in Table 1.

Due to lack of space, graphical results are not shown here but can be obtained using the analysis in section 3. From these graphical results, the data transmission efficiency, ϕ, decreases as the maximum number of devices increases. It does not depend

Table 1. Parameter Values Used

Symbol	Value	Symbol	Value
N_{SF}	64	$1/\mu_1$	1 minute
N_{BS}	3	$1/\mu_2$	2 minutes
N_{BC}	3	α_1	0.352
M_1	1	β_1	0.650
M_2	2	α_2	0.9
r_1	1	β_2	0.1
r_2	2	P_B^*	10^{-3}
λ_1	λ_2	P_L^*	10^{-3}
λ	$\lambda_1 + \lambda_2$	ϕ_e^*	0.8

on the total device mean connection arrival rate but only on the total number of devices in the system. If we choose the data transmission efficiency requirement, ϕ^*, to be greater than 0.8, then the maximum number of devices, N_D, that can be supported is only 33. The admission criterion is simply limited to no more than 33 devices. The utilization N_u={22.3,23.3,22.6,21.8,21.0} when the total device mean connection arrival rate, $\lambda(=\lambda_1+\lambda_2)$, is {10,20,30,40,50} arrivals per minute.

Similarly from the graphical results, the effective data transmission efficiency, ϕ_e, decreases at a slower rate compared to the data transmission efficiency, ϕ, as the maximum number of devices increases. However, it decreases as the total device mean connection arrival rate increases. Solving the cross-layer optimization in Section 3, we have the maximum utilization, N_u={36.1,46.1,46.8,46.2,45.3} packet transmission slots at the maximum number of devices, N_D={65,63,63,63,63} for the total device mean connection arrival rate, λ={10,20,30,40,50} arrivals per minute. The improvement in utilization are respectively {62%,98%,107%,112%,116%}. Thus this optimized solution results in much higher utilization compared to the case where the data transmission efficiency requirement, ϕ^*, is chosen to be greater than 0.8. The limiting constraint in this numerical example is caused by the packet loss probability requirement, P_L^*, at 10^{-3}. Note that the maximum number of devices, N_D, to achieve maximum utilization, N_u, is quite insensitive to the total device mean connection arrival rate. Thus, the maximum number of devices, N_D, can be set at 63, for example, where the utilization is maximized for most of the total device mean connection arrival rate under consideration. This is a simple admission criterion. In practical systems, measured average device arrival rates are needed.

5 Concluding Remarks

A cross-layer optimization problem has been formulated to maximize utilization in a distributed real-time MAC protocol for WPANs. The GoS/QoSs performance metrics in the connection level in the network layer, the packet level in the network layer and the MAC layer have been coupled together to optimize system performance. Numerical results show that this cross-layer optimization approach results in much higher utilization (62% to 116%) than the approach that simply considering the data transmission efficiency requirement. The analysis here can be used to determine the optimal maximum number of devices that can be admitted into the system such that the utilization is maximized under different total device mean connection arrival rate in a distributed real-time MAC protocol for WPANs.

References

1. Wijting, C., Prasad, R.: A Generic Framework for Cross-Layer Optimisation in Wireless Personal Area Networks. Wireless Personal Communications, Vol 29, (2004) 135-149
2. O'Conor, J., Brown, R.: MBOA Technical Specification: Distributed Medium Access Control (MAC) for Wireless Networks. MBOA Draft MAC standard version 0.95, (11 April 2005)
3. Maglaris, B., Anastassiou, D., Sen, P., Karlsson, G., Robbins, J.D.: Performance Models of Statistical Multiplexing in Packet Video Communications. IEEE Transactions on Communications, Vol. 36, No. 7, (July 1988) 834-844

Performance Analysis of IEEE802.16e Random Access Protocol with Mobility

Sang-Sik Ahn[1], Hyong-Woo Lee[1], Jun-Bae Seo[2], and Choong-Ho Cho[3]

[1] Department of Electronics and Information Engineering,
Korea University
sahn@korea.ac.kr, hwlee@korea.ac.kr
[2] Department of Wireless System Research,
Electronics and Telecommunications Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeong 305-350, South-Korea
jbseo@etri.re.kr
[3] Department of Computer Science, Korea University,
Chochiwon, Chungnam 339-700, South-Korea
chcho@korea.ac.kr

Abstract. In this paper, the performance of IEEE802.16e random access protocol with handover procedure is examined in terms of access throughput and mean access delay, by using equilibrium point analysis(EPA). In the analysis, retransmission probability, which is a typical input parameter in the literature so far, is iteratively obtained from equilibrium number of backlogs in the system in conjunction with a binary exponential backoff algorithm. In numerical examples, the effects of SSs' mobility on access throughput and mean access delay are examined.

1 Introduction

Among the various features of the physical layer in IEEE802.16a/b/c/d/e, we focus on orthogonal frequency-division-multiplexing(OFDM) with TDD mode. The frame structure and its detailed description of our interest are given in [1]-[3]. In this paper, we examine the performance of IEEE 802.16e MAC protocol with mobility by using EPA, since the analysis using a Markov chain to describe various states of a subscriber station(SS) is formidable [3] due to the explosion of the state space. One can find some previous works on IEEE802.16 random access protocol [3]-[6] in the literature. This paper is organized as follows. In section 2, the handover procedure of IEEE802.16e MAC protocol is described and its analysis is given. The numerical examples are discussed in section 3. Concluding remarks are given in section 4.

2 IEEE 802.16 MAC Protocol

2.1 Procedure of Bandwidth Request and Handover Ranging

The random access protocol of IEEE802.16e is a class of demand-assigned multiple access(DAMA). The basic procedure of the random access protocol without handover

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1106–1112, 2006.

procedure is summarized in [3]. Here, we focus on the handover procedure(HO) only. Whenever an SS crosses a cell-boundary irrespective of its actions, such as, retransmissions of bandwidth request code, data transmission or waiting for CDMA allocation message, and so on, it performs the following handover procedure. According to the carrier-to-interference-noise ratio(CINR) of a serving base station(BS), an SS sends HO-request message and receives HO-response message. After that, the SS sends HO-indication message to the serving BS. At this point, the SS doesn't scan DL/UL-MAPs of the serving BS any more. Note that HO-request and HO-indication messages from the SS are also delivered through the random access procedure in parallel with data traffic transmission. Here, we assume that these two signalling messages are negligible for our modelling, because of their parallel transmission structure. In a target BS, the SS performs a HO-ranging procedure which is contention-based synchronization(or adjustment) to the system. It uses a PN code in HO ranging code group. Note that an SS has already known a group of HO ranging codes in a target BS from the neighbor cell advertisement message in the previous serving cell. At the end of HO-ranging procedure, registration, authentication and other procedures may follow or be omitted according to the HO optimization field within HO ranging response message. In case of an omission of such signalling procedures, the information of an SS is transferred to the target BS from the serving BS. The detailed operations are given in [2]. In order to complete HO procedures, the SS can restart the bandwidth ranging procedure. Therefore, the time before restarting the bandwidth ranging procedure in a target BS from the transmission termination in the previous serving BS forms a random delay at least due to the HO ranging procedure, if we may view the initiation epoch of HO as the beginning of HO-indication message transmission. For simplicity of the analysis, we assume that the elapsed time up to the bandwidth ranging procedure in a target BS from the HO ranging procedure, which includes signalling delay, takes y frames.

2.2 Equilibrium Point Analysis

In TDD mode, it is hard to transmit a response message on DL-subframe in the $(i+1)$-th frame when the message corresponding to the response message has been received on UL-subframe in the i-th frame, because the decoding time for the message received from UL-subframe and the encoding time for the message to transmit on DL-subframe may overlap for practical implementation. We assume that the delay from the reception of the bandwidth request code to transmission of its response, i.e., processing delay, and the delay from the reception of the bandwidth request message to transmission of its channel allocation, i.e., scheduling delay, are respectively z and x frames. In addition, the delay by **T3** timer on retransmissions and the delay of the HO procedure are respectively assumed to be w and y frames.

In Fig.1, a set of modes an SS can be in is shown. The modes C, B, R and T respectively denote the initial bandwidth request code transmission-, its retransmissions-, bandwidth request message transmission- and data transmission-mode. Additionally, P_{Ti}, P_{Bj}, P_{Rk} and for $0 \leq i \leq z-1$, $0 \leq j \leq x-1$ and $0 \leq k \leq w-1$, denote the delay experienced by C, R and B mode. The modes of H and $\underline{\mathrm{H}}_l$ for $0 \leq l \leq y-1$ denote the HO ranging procedure with transition probability, p_h, and signalling

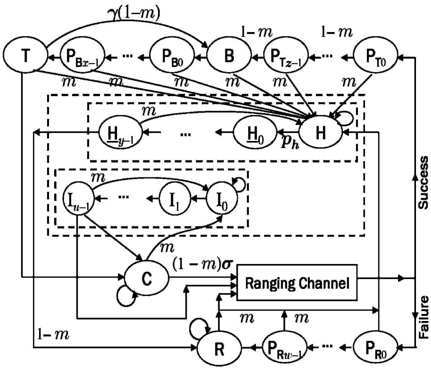


Fig. 1. A model of IEEE802.16e MAC protocol

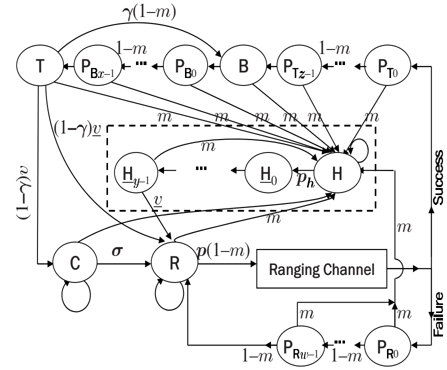


Fig. 2. A modified model of IEEE802.16e MAC protocol

delay. Finally, the mode, I_n, for $0 \leq n \leq u-1$, denotes that an SS moves out of a serving BS between traffic arrivals. Transition from one mode to another occurs at the end of a frame. In mode C, an SS has data to be transmitted with probability σ. In mode R, an SS retransmits the bandwidth request code with probability p, after it is known that the previous transmission of a bandwidth request code is not successful. The transition from T to B indicates the piggyback of the bandwidth request message at the end of data transmission with probability γ. In each mode, it may move out of a serving BS with probability m. Once an SS enters the mode of $\underline{\mathrm{H}}_0$, its transitions to $\underline{\mathrm{H}}_l$ for $1 \leq l \leq y-1$ occur with probability one. We assume a finite population of M SSs in a given cell. Before proceeding further, we modify the model in Fig.1 under the condition of $\sigma \leq p$ as shown in Fig.2 in order to merge two inputs to the ranging channel into one[7]. In order to focus on the effect of mobility upon traffic transmission, the mode, I_n, $\forall n$, are ignored. The parameters, v and $\underline{v}$, in Fig.2 are expressed as $v{=}\widehat{m}(1{-}\sigma/p)$, $\underline{v}{=}\widehat{m}(\sigma/p)$ and $\widehat{m}{=}1{-}m$. Accordingly, in a given BS, the system state is described by the vector (C, R, B, T, P_{Ti}, P_{Bj}, P_{Rk}, H, $\underline{H}_l$), $\forall i$, $\forall j$, $\forall k$ and $\forall l$, where C is the number of SSs in mode C, R is the number of SSs in mode R, and so on. We denote the equilibrium variables of the state variables for the system by the corresponding lowercase letters, (c, r, b, t, p_{Ti}, p_{Bj}, p_{Rk}, h, $\underline{h}_l$), $\forall i$, $\forall j$, $\forall k$ and $\forall l$. Since the transitions within the states to represent the delay, p_{Ti}, p_{Bj}, p_{Rk} $\forall i$, $\forall j$ and $\forall k$, occur with probability $1-m$, one can readily find the following relations : $p_{Ti+1}{=}\widehat{m}p_{Ti}{=}\widehat{m}^{i+1}p_{T0}$, $p_{Bj+1}{=}\widehat{m}^{j+1}p_{B0}$ and $p_{Rk+1}{=}\widehat{m}^{k+1}p_{R0}$, $\forall i$, $\forall j$, $\forall k$. At the states, P_{T0}, P_{R0}, P_{B0}, B and T, one can also obtain the followings : $p_{T0}{=}S(r)$, $p_{R0}{=}p\widehat{m}r{-}S(r)$, $p_{B0}{=}\widehat{m}b$, $b{=}\widehat{m}\big(p_{Tz-1}{+}\gamma t\big)$ and $t{=}\widehat{m}p_{Bx-1}$, where $S(r)$ is the expected input to the system, which will be derived later. By the same way, at the states, $\underline{H}_l$ and H, one can get the followings : $\underline{h}_{l+1}{=}\underline{h}_l$, $\forall l$, $\underline{h}_0{=}p_h h$ and $h{=}m\big(\underline{h}_{y-1}{+}r{+}b{+}t{+}\check{d}\big){+}(1-p_h)h$, where $\check{d}{=}\sum_{i=0}^{z-1} p_{Ti}{+}\sum_{j=0}^{x-1} p_{Bj}{+}\sum_{k=0}^{w-1} p_{Rk}$. Because the sum of SSs in all the states must be M, the following equation is satisfied.

$$(1-\gamma)vt = \sigma\Big(M - r - t - b - \check{d} - \sum_{l=0}^{y-1} \underline{h}_l - h\Big) \tag{1}$$

With some manipulations, one can express b, t and each sum of p_{Ti}, p_{Bj} and p_{Rk}, $\forall i$, $\forall j$ and $\forall k$, in terms of r and $S(r)$. Substituting these into (1) and rearranging it as

$$M = \left[(1+\mu)\left(1 + \frac{\widehat{m}}{m} p(1-\widehat{m}^w)\right)\right] r + \left[\left[\frac{(1+\gamma)v+\sigma}{\sigma} + \mu\right]\widehat{m}^x \phi \right.$$
$$\left. + (1+\mu)\left[\phi + m^{-1}\Big(\widehat{m}^w + (1-\widehat{m}^x)\widehat{m}\phi - \widehat{m}^z\Big)\right]\right] S(r) \tag{2}$$

in which $\phi = \widehat{m}^z/(1-\gamma\widehat{m}^{x+1})$. The expected input to the system, $S(r)$, can be expressed as $S(r) = L_t f(r)$, where $f(r)$ indicates the mean number of successfully received PN-codes on a slot-subchannel given r SSs.

$$f(r) = \sum_{k=0}^{r}\sum_{j=0}^{k} P_d(j)\phi_c(j;N_c)\binom{k}{j}(1/L_t)^j(1-1/L_t)^{k-j}\binom{r}{k}\widetilde{p}^k(1-\widetilde{p})^{r-k} \tag{3}$$

with $\widetilde{p} = p(1-m)$. We denote the mean number of the distinct PN codes transmitted on a slot-subchannel, provided that j SSs transmit PN codes randomly chosen among total of N_c PN codes by $\phi_c(j;N_c)$, which is given in [3]. $P_d(j)$ is the probability that a PN code sent by an SS will be successfully identified among j codes in a slot-subchannel given N_b neighboring BSs, which is heuristically expressed as $P_d(j) = \left(e^{\theta(j-\theta_t)} + \varepsilon\sqrt{N_b}\right)^{-1}$, where the first and second terms in the right-hand side represent intra- and inter-cell multiple access interference(MAI), respectively. The parameter, θ, is the extent of degradation due to MAI and θ_t is threshold at which $P_d(j)$ is 1/2, when ε=N_b=1. The parameter, ε, is a weight for intercell MAI. The second term must be greater than or equal to 1. $P_d(j)$ can be also derived by considering physical parameters[8]. For measuring performance, we define r_e as the equilibrium number of r in R state of the system which can be obtained by solving (2). Using Little's result, one can obtain the initial access delay, $\overline{D}$, and access throughput, $\overline{S}$, respectively as follows.

$$\overline{D} = \Big(r_e + \sum_{k=0}^{w-1} p_{Rk} + \sum_{l=0}^{y-1} \underline{h}_l + h\Big)/\overline{S} \tag{4}$$

with $\overline{S} = S(r_e)$. Whenever unsuccessful accesses occur, each SS involved increases its contention window size in a binary exponential manner. That is, the contention window size after the k-th collision, $W(k)$, is given by

$$W(k) = \min\Big(W_0 2^{(k-1)}, W_m\Big), \quad 1 \le k \le K_m \tag{5}$$

where W_0 is an initial window size and W_m is the maximum of $W(k)$. When $W(k)$ reaches W_m, W_m is repeatedly used. An SS deferes its retransmission the time randomly selected among $W(k)$. In order to include a binary exponential backoff algorithm, we estimate the retransmission probability, p, as follows. With an initial guess for p and r_e, one can obtain the transmission success probability, p_s, as follows.

$$\overline{W} = \sum_{k=1}^{K_m} p_s(1-p_s)^k \left(\frac{W(k)-1}{2}\right), \quad \text{with} \quad p_s = S(r_e)/r_e \tag{6}$$

where K_m is the maximum exponent of $W(k)$. Then, the retransmission probability, p, can be obtained by $p{=}1/\overline{W}$. By updating, and substituting p_s and p into (2) at each iteration, we can obtain $\overline{S}$ and $\overline{D}$ when both p and r_e converge.

3 Numerical Examples

The parameters of $P_d(j)$, $\theta{=}1.5$, $\theta_t{=}4.5$, $\varepsilon{=}0.42$ and $N_b{=}6$ are used. For delay parameters, $x{=}3$, $y{=}8$, $z{=}3$ and $w{=}5$ are also used with $p_h{=}0.8$. Additionally, the parameters of the binary exponential backoff algorithm, $W_0{=}1$, $N_m{=}6$ and $W_m{=}70$, are used. Finally, the piggyback probability, γ, number of PN codes, N_c, number of slot-subchannels, L_t, and the traffic generation probability, σ, are respectively set to 0.01, 4 and 6.

In Figs.3 and 4, by varying HO occurrence probability, m, HO ranging success probability, p_h, and the population size, M, $\overline{S}$ and $\overline{D}$ are respectively depicted. When $m{=}0$,

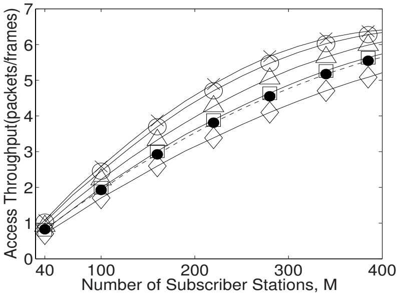


Fig. 3. Access Throughput($y = 8$, $m =$ [×]0.001, [○]0.01, [△]0.03, [□]0.05, [◇] 0.08, [●] $y = 5$, $m = 0.08$)

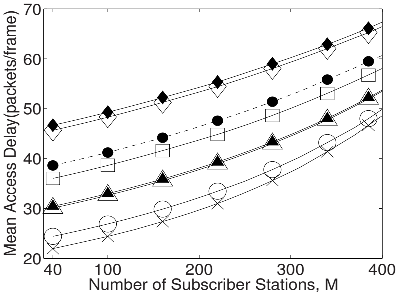


Fig. 4. Mean Access Delay($y = 8$, $p_h = 0.8$, $m =$ [×]0.001, [○]0.01, [△]0.03, [□]0.05, [◇] 0.08, [●] $y = 5$, $m = 0.08$; $p_h = 0.6$, $m =$ [▲] 0.03 , [♦] 0.08)

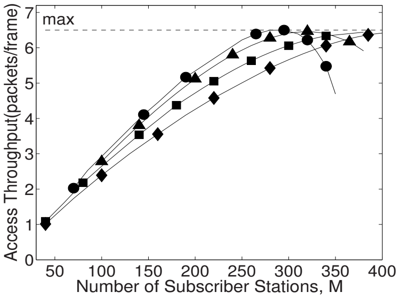


Fig. 5. Access Throughput(m=0.01, σ=0.04, W_0=5, W_m=[●]8, [▲]16, [■]32, [♦] 64)

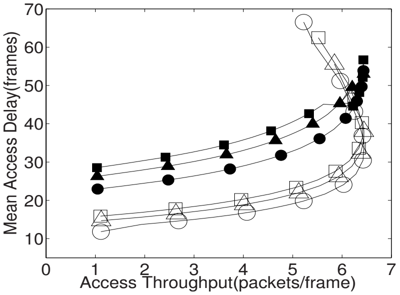


Fig. 6. Mean Access Delay vs. Access Throughput (m=0.01, σ=0.04; W_m=32, W_0= [○]1, [△]3, [□]5; W_m=64, W_0=[●]1, [▲]3, [■]5)

the model just considers the performance of IEEE 802.16d random access protocol, which doesn't include handover procedure. When m becomes large, $\overline{S}$ and $\overline{D}$ respectively decreases and increases. This can be expected, because SSs with high mobility will more frequently experience the handover procedure which includes its own delay and the retransmission procedure, compared to SSs with low mobility. In addition, when handover signalling delay, y, is reduced, $\overline{S}$ and $\overline{D}$ respectively increases and decreases. In Figs.5, $\overline{S}$ is observed according to W_m. When W_m increases, the maximum of $\overline{S}$ moves toward large M. In Fig.6, $\overline{S}$ is depicted versus $\overline{D}$.

4 Conclusion

In this paper, we examined the performance of random access protocol of IEEE802. 16e with handover procedure, in terms of access throughput and mean access delay as an extention of [3]. The increase of mobility results in the reduction of access throughput and the increase of mean access delay, which results from the fact that each handover process includes the random access procedure as well as its own signalling delay. Although EPA provides accurate results for a stable system with large number of stationary SSs in a BS [7], as a future work, the analytical results may be validated by simulation according to handover occurrence probability, due to variance of number of SSs handed over among BSs. It would be also interesting to analyze the performance when an SS's mobility shows memory.

Acknowledgement

This work was supported by grant No.B1220-0501-0232(2005) from the University fundamental Research Program of the Ministry of Information & Communication in Republic of Korea.

References

1. Draft IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems(IEEE802.16 REVd/D5-2004), May, 2004.
2. Draft IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems-Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands(IEEE P802.16e/D7), April, 2005.
3. Jun-Bae Seo, Nam-Suk Lee, Nam-Hoon Park, Hyong-Woo Lee and Choong-Ho Cho, "Performance Analysis of IEEE802.16d Random Access Protocol," *Accepted for publication at IEEE ICC'06, Available on Request*.
4. Jeong-Jae Won, Choong-Ho Cho, Hyong-Woo Lee and Victor Leung, "Stabilization of Contention-Based CDMA Ranging channel in Wireless Metropolitan Area Networks," *in Proc. of Networking 2005*, LNCS vol.3462, pp.1255-1266, Waterloo, Canada, May 2-6, 2005.
5. Hyun-Hwa Seo, Byung-Han Ryu, Hyong-Woo Lee and Choong-Ho Cho, "Design of Performance Analysis Model for Efficient Random Access Protocol in CDMA based OFDMA-PHY System," *IEEE ICACT'05*, vol.1, pp.347-351, 2005.

6. Hyun-Hwa Seo, Choong-Ho Cho, Hyong-Woo Lee, "Traffic Characteristics based Performance Analysis Model for Efficient Random Access in OFDMA-PHY System," *in Proc. of WWIC 2005*, LNCS vol.3510, pp.213-222, Xanthi, Greece, May 11-13, 2005.
7. Shuji Tasaka, *Performance Analysis of Multiple Access Protocols*, MIT Press, 1986.
8. Jisang You, Kanghee Kim and Kiseon Kim, "Capacity evaluation of the OFDMA-CDMA ranging subsystem in IEEE802.16-2004," *in Proc. of IEEE WiMob'05*, vol.1, pp.100-106.

Cost-Benefit Analysis of Web Prefetching Algorithms from the User's Point of View*

Josep Domènech, Ana Pont, Julio Sahuquillo, and José A. Gil

Department of Computing Engineering (DISCA),
Universitat Politècnica de València, Spain
jodode@doctor.upv.es, {apont, jsahuqui, jagil}@disca.upv.es

Abstract. Since web prefetching techniques were proposed in the second half of the 90s as mechanisms to reduce final users' perceived latency, few attempts to evaluate their performance have been done in the research literature. Even more, to the knowledge of the authors this is the first study that evaluates different proposals from the user's point of view, i.e., considering the latency perceived by the user as the key metric. This gap between the proposals and their correct performance comparison is due to the difficulty to use a homogeneous framework and workload. This paper is aimed at reducing this gap by proposing a cost-benefit analysis methodology to fairly compare prefetching algorithms from the user's point of view. The proposed methodology has been used to compare three of the most used algorithms in the bibliography, considering current workloads.

1 Introduction

Several ways of prefetching user's requests have been proposed in the literature: the preprocessing of a request by the server [1], the transference of the object requested in advance [2], and the pre-establishment of connections that are predicted to be made [3]. Despite the large amount of research works focusing on this topic, comparative and evaluation studies from the user's point of view are rare. On the one hand, the underlying baseline system where prefetching is applied differs widely among the studies. On the other hand, different performance key metrics were used to evaluate their benefits [4]. In addition, the used workloads are in most cases rather old, which significantly affects the prefetching performance [5], making the conclusions not valid for current workloads.

Researchers usually compare the proposed prefetching system with a non-prefetching one [6, 2], under heterogeneous conditions making it impossible to compare the goodness and benefits of each proposal.

Some papers comparing the performance of prefetching algorithms have been published [7, 8, 9, 10, 11] but they mainly concentrate on predictive performance [7, 8, 9, 10].

* This work has been partially supported by Spanish Ministry of Education and Science and the European Investment Fund for Regional Development (FEDER) under grant TSI 2005-07876-C03-01.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1113–1118, 2006.

In addition, performance comparisons are rarely made using a useful cost-benefit analysis, i.e., latency reduction as a function of the traffic increase. As examples of some timid attempts, Dongshan and Junyi [7] compare the accuracy, the model-building time, and the prediction time in three versions of a predictor based in Markov chains. Another current work by Chen and Zhang [8] implements three variants of the PPM predictor by measuring the hit ratio and traffic under different assumptions.

Nanopoulos *et al.* [9] show a cost-benefit analysis of the performance of four prediction algorithms by comparing the precision and the recall to the traffic increase. Nevertheless, they ignore how the prediction performance affects the final user. Bouras *et al.* in [10] show the performance achieved by two configurations of the PPM algorithm and three of the n-most popular algorithm. They quantify the usefulness (recall), the hit ratio (precision) and the traffic increase but they present a low number of experiments, which make it difficult to obtain conclusions. In a more recent work [11] they also show an estimated upper bound of the latency reduction for the same experiments.

In this paper we propose and implement a cost-benefit methodology to perform fair comparisons of web prefetching algorithms from the user's point of view. Some experiments were performed to illustrate how we can evaluate the benefits of the prefetching.

The remainder of this paper is organized as follows. Section 2 describes the experimental environment used to run the experiments. Section 3 proposes a methodology to evaluate prefetching algorithms. Section 4 analyzes the experimental results of an example of application of the proposed methodology. Finally, Section 5 presents some concluding remarks.

2 Experimental Environment

2.1 Framework

In [12] we proposed an experimental framework for testing web prefetching techniques. In this section we summarize the main features of such environment and the configuration used to carry out the experiments presented in this paper.

The architecture consists of two main parts: the back end (server and surrogate), and the front end (client). The framework implementation combines both real and simulated parts in order to provide flexibility and accuracy.

The back end part includes the web server and the surrogate server. The framework emulates a real surrogate, which is used to access a real web server. We use the surrogate as a predictor. To this end, it adds new HTTP headers to the server response with the result of the prediction algorithms, as implemented in Mozilla. The server is an Apache web server set up to act as the original one. For this purpose, it has been developed a CGI program that returns objects with the same size and MIME type than those recorded in the traces.

The front end, or client part, represents the users' behavior exploring the Web with a prefetching enabled browser. To model the set of users that access concurrently to a given server, the simulator is fed by using real traces. The simulator collects basic information for each request performed to the web server, then writes it to a log file. By analyzing this log at post-simulation time, all performance metrics can be calculated.

2.2 Workload Description

The behavior pattern of users was taken from two different logs. Traces A and B were collected during May 12^{th} 2003. They were obtained by filtering their accesses in the log of a Squid proxy of the Polytechnic University of Valencia. The trace A contains accesses to a news web server, whereas the trace B has the accesses to a student information web server. The main characteristics of the traces are shown in Table 1. The training length of each trace has been adjusted to optimize the perceived latency reduction of the prefetching.

Table 1. Traces characteristics

Characteristics	**Trace**	
	A	**B**
Year	2003	2003
Users	300	132
Page Accesses	2,263	1,646
Objects Accesses	65,569	36,837
Training length (accesses)	35,000	5,000
Bytes Transferred (MB)	218.09	142.49

2.3 Prefetching Algorithms

The experiments were run using three of the most widely used prediction algorithms in the literature: two main variants of the *Prediction by Partial Match* (PPM) algorithm [13, 7, 8, 9] and the *Dependency Graph* (DG) based algorithm [2, 9].

The PPM prediction algorithm uses Markov models of m orders to store previous contexts. Predictions are obtained from the comparison of the current context to each Markov model. PPM algorithm has been proposed to be applied either to each object access [13] or to each page (i.e., to each container object) accessed by the user [7, 8]. In this paper we implement the object-based version of the algorithm.

The DG prediction algorithm constructs a weighted dependency graph that depicts the pattern of accesses to the objects. The prefetching aggressiveness is controlled by a cutoff threshold parameter applied to the arcs weight.

2.4 Performance Indexes

The performance of the algorithms has been evaluated using the two main user related metrics [4], each one representing the cost and the benefit of the web prefetching. Both indexes are better as lower their value is.

- Latency per page ratio (L_p): The latency per page ratio is the ratio of the latency that prefetching achieves to the latency with no prefetching. The latency per page is calculated by comparing the time between the browser initiation of an HTML page GET and the browser reception of the last byte of the last embedded image or object for that page.

- Traffic Increase (ΔTr): The bytes transferred through the network when prefetching is employed divided by the bytes transferred in the non-prefetching case. Notice that this metric includes both the extra bytes wasted by prefetched objects that the user will never use, and the network overhead caused by the transference of the prefetch hints.

3 Methodology

The comparison of prefetching algorithms should be made from the user's point of view and using a cost-benefit analysis. Despite the fact that prefetching has been also used to reduce the peaks of bandwidth demand [14], its primary goal; i.e., the benefit, is usually the reduction of the user's perceived latency.

When predictions fail, prefetched objects waste user and/or server resources. Since in most proposals the client downloads the predicted objects in advance, the main cost of the latency reduction in prefetching systems is the network traffic increase. As a consequence, the performance analysis should consider the benefit of reducing the user's perceived latency at the cost of increasing the network traffic.

For comparison purposes, we have simulated systems implementing the above described algorithms. Each simulation experiment on a prefetching system takes as input the user behaviour and the prefetching parameters. The main results obtained are the traffic increase and the latency per page ratio values.

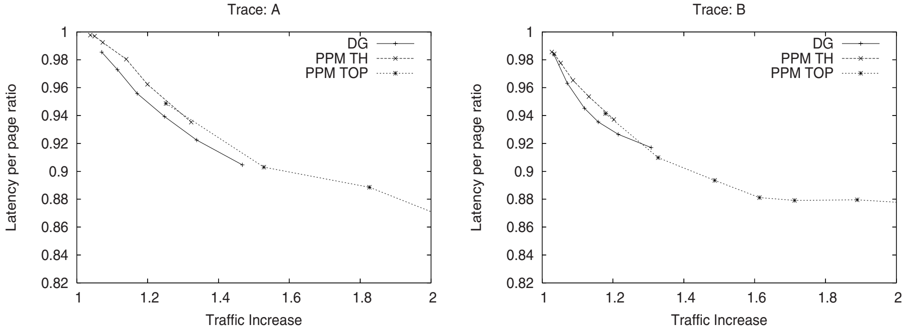
Comparisons of two different algorithms only can be fairly done if either the benefit or the cost have the same or close value. For instance, when two algorithms present the same or very close values of traffic increase, the best proposal is the one that presents less user perceived latency, and vice versa.

For this reason, in the examples shown in this paper the performance comparisons are made through curves that include different pairs of traffic increase and latency per page ratio for each algorithm. In order to obtain each point in the curve we have varied the aggressiveness of the algorithm, i.e., how much an algorithm will predict. This aggressiveness is controlled by a threshold parameter in those algorithms that support it (i.e., DG and PPM-TH) and by the number of returned predictions in the PPM-TOP.

A plot can gather the curves obtained for each algorithm in order to be compared. By drawing a line over the desired latency reduction in this plot, one can obtain the traffic increase of each algorithm. The best algorithm for achieving that latency per page is the one having less traffic increase.

4 Algorithms Comparison

Figure 1 shows the results for the algorithms described in Sect. 2.3. Each algorithm is evaluated in two situations each one using one of the two described



(a) Trace A

(b) Trace B

Fig. 1. Performance comparison between objects based algorithms. Each point in the curves represents a given threshold in PPM-TH and DG, while it represents a given amount of returned hints in PPM-TOP.

workloads (i.e., A and B). The curves of each plot in DG and PPM-TH algorithms are obtained by varying the confidence threshold of the algorithms, from 0.2 to 0.7 in steps of 0.1. To make the curves of the PPM-TOP algorithm, the number of returned predictions are ranged from 1 to 9 in steps of 1, except for 6 and 8. Results for traffic increases greater than 2 are not represented in order to keep the plot focused on the area where the algorithms can be compared.

Figure 1(a) illustrates the performance evaluation of the algorithms simulating users who have 1 Mbps of available bandwidth and behave in accordance with the workload A. This plot shows that the DG algorithm achieves better performance than the others in the range in which it is evaluated, since its curve falls always below the ones of the PPM algorithms.

Figure 1(b) shows that the algorithms exhibit minor performance differences when using the trace B. DG algorithm slightly outperforms the others in all its range with the only exception of the most aggressive threshold (i.e., *th*=0.2), in which the PPM-TOP algorithm achieves a slightly higher latency reduction with the same traffic increase.

5 Conclusions

A large amount of research works has focused on web prefetching. However, comparative studies are rare and usually ignore the user's point of view. In this paper we have described a cost-benefit methodology to evaluate and compare prefetching algorithms from the user's point of view.

Using the proposed methodology, three prediction algorithms have been implemented and compared. Experimental results show that DG algorithm slightly outperforms the PPM-TH and the PPM-TOP algorithms in most of the analyzed cases. However, the aggressiveness (and, consequently, the latency reduction) of the DG is more limited than the PPM-TOP one. For this reason, when prefetching is not desired to be very aggressive, DG achieves the best cost-effectiveness.

References

[1] Schechter, S., Krishnan, M., Smith, M.D.: Using path profiles to predict http requests. In: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia (1998)

[2] Padmanabhan, V.N., Mogul, J.C.: Using predictive prefetching to improve World-Wide Web latency. In: Proceedings of the ACM SIGCOMM '96 Conference, Stanford University, USA (1996)

[3] Cohen, E., Kaplan, H.: Prefetching the means for document transfer: a new approach for reducing web latency. Computer Networks **39** (2002)

[4] Domènech, J., Gil, J.A., Sahuquillo, J., Pont, A.: Web prefetching performance metrics: A survey. Accepted to be published in Performance Evaluation (2006)

[5] Domènech, J., Sahuquillo, J., Pont, A., Gil, J.A.: How current web generation affects prediction algorithms performance. In: Proceedings of SoftCOM Int. Conf. on Software, Telecommunications and Computer Networks, Split, Croatia (2005)

[6] Duchamp, D.: Prefetching hyperlinks. In: Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems, Boulder, USA (1999)

[7] Dongshan, X., Junyi, S.: A new markov model for web access prediction. Computing in Science and Engineering **4** (2002)

[8] Chen, X., Zhang, X.: A popularity-based prediction model for web prefetching. IEEE Computer **36** (2003)

[9] Nanopoulos, A., Katsaros, D., Manolopoulos, Y.: A data mining algorithm for generalized web prefetching. IEEE Trans. Knowl. Data Eng. **15** (2003)

[10] Bouras, C., Konidaris, A., Kostoulas, D.: Efficient reduction of web latency through predictive prefetching on a wan. In: Proceedings of the 4th Int. Conf. on Advances in Web-Age Information Management, Chengdu, China (2003)

[11] Bouras, C., Konidaris, A., Kostoulas, D.: Predictive prefetching on the web and its potential impact in the wide area. World Wide Web **7** (2004)

[12] Domènech, J., Pont, A., Sahuquillo, J., Gil, J.A.: An experimental framework for testing web prefetching techniques. In: Proceedings of the 30th EUROMICRO Conference 2004, Rennes, France (2004)

[13] Sarukkai, R.: Link prediction and path analysis using markov chains. Computer Networks **33** (2000)

[14] Maltzahn, C., Richardson, K.J., Grunwald, D., Martin, J.H.: On bandwidth smoothing. In: Proceedings of the 4th International Web Caching Workshop, San Diego, USA (1999)

An MPLS-Based Micro-mobility Solution

IEEE-802.21-Based Control Plane

Rajendra Persaud[1], Ralf Wienzek[1], Gerald Berghoff[2], and Ralf Schanko[2]

[1] Chair of Computer Science 4, RWTH Aachen University, Germany
[2] Nokia Networks GmbH, Germany

Abstract. Core network micro-mobility solutions resolve L3 handovers and may be based on the Internet Protocol (IP) or on Multi-Protocol Label Switching (MPLS). When a micro-mobility solution triggers the L3 handover before the L2 handover, it is called predictive, otherwise reactive. The outage period due to the handover is smaller for predictive solutions. However, in order to be predictive, a L3 mobility solution needs support from the underlying link-layer. This support may be provided with the help of IEEE 802.21 that is exploited for intra-technology handovers in WLANs in this paper.

1 Introduction

Each wireless network may be subdivided into an access network and a core network. A mobile device is generally attached with a link-layer (L2) Point of Attachment (PoA) in the access network and a network-layer (L3) PoA in the core network. A handover between two L2 PoAs belonging to the same access network is generally resolved by a L2 mobility solution and called L2 handover. A handover between two L2 PoAs belonging to different access networks is generally resolved by a L3 mobility solution and called L3 handover. Note that a L3 handover includes a L2 handover. The focus of this paper is on L3 handovers.

Such a L3 handover is implemented by a L3 mobility solution which may either be based on the Internet Protocol (IP) or on Multi-Protocol Label Switching (MPLS). The focus of this paper is on MPLS-based intra-domain (i.e. micro-mobility) solutions. Any L3 mobility solution can be viewed as either predictive or reactive. A predictive solution starts the L3 handover before the L2 handover, a reactive solution starts the L3 handover after the L2 handover. The great advantage of a predictive solution is that L2 and L3 handovers can be executed in parallel and not in sequence as in reactive solutions. The outage period in predictive solutions is thus generally lower than in reactive solutions.

In order to help with the decision when to start the L3 handover, the mechanisms of IEEE 802.21 can be used. IEEE 802.21 has been conceived as L2/L3 management layer in order to support L3 mobility solutions for inter-technology handovers. Whenever a technology does not incorporate a mechanism for higher-layer mobility, which is the case for WLANs, IEEE 802.21 can be exploited for intra-technology handovers as well.

IEEE 802.21 provides an Event Service, a Command Service and an Information Service by a Media Independent Handover Function (MIHF). The MIHF

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1119–1124, 2006.

transfers higher-layer commands into corresponding L2 commands and L2 events into corresponding higher-layer events. Although IEEE 802.21 is still in draft status, the basic idea and the basic events, commands and messages are already specified and shall be exploited in this paper.

Note that a handover is primarily an issue for downstream data packets. For upstream data packets, no location updates or redirections of data packets are necessary in the core network.

2 Previous Work

Most IP-based micro-mobility solutions such as Hierarchical Handovers for Mobile IPv6 (HMIPv6) are reactive solutions. The reason is that the mobile device needs to have a new IP address at the new L3 PoA, which it can, in general, only acquire at the new L3 PoA belonging to the new subnet. With IPv6 Stateless Address Autoconfiguration this restriction is removed for IPv6 and exploited by the predictive mode of Fast Handovers for Mobile IPv6 (FMIPv6).

Many MPLS-based mobility management solutions such as [1, 2, 3] propose a combination of MPLS and Mobile IPv4 or of MPLS and HMIPv6. While MIPv4 and HMIPv6 are exploited to provide the signalling necessary for mobility management, MPLS is exploited instead of IP encapsulation for data delivery on the user plane. The main objective is thus to reduce the overhead of IP encapsulation. Since the cited mobility solutions are based on the signalling of MIPv4 and HMIPv6, they are all reactive solutions.

A purely MPLS-based approach introducing the concept of Label Edge Mobility Agents (LEMAs) is presented by [4]. A LEMA is an LER enhanced by a function for mobility management. The objective of deploying LEMAs is to reduce the time needed for the location update at the domain ingress router. The LEMA approach is a reactive MPLS-based micro-mobility solution.

In [5], another purely MPLS-based handover solution is proposed. Before a handover, the traffic of a particular user is sent as part of an aggregated traffic flow over a primary Label Switched Path (LSP) to the previous L3 PoA. During the handover, the user traffic is separated from the aggregated traffic flow and successively placed on handover LSPs leading towards the new L3 PoA. After the handover, the user traffic is re-inserted into an aggregated traffic flow, this time towards the new L3 PoA.

3 Start-Up

The main objective of the start-up procedure is to enable the mobile device to send and receive IP packets. Therefore, the mobile device has to be registered in the core network. If IEEE 802.11 is the underlying L2 technology, the mobile device authenticates and associates with an Access Point (AP) (cf. Fig. 1 (a)). If IEEE 802.11i is used, the AP has to be capable of using an Authentication, Authorization and Accounting (AAA) protocol such as Remote Dial-In User Service (RADIUS). Since the AP located in the access network would have to

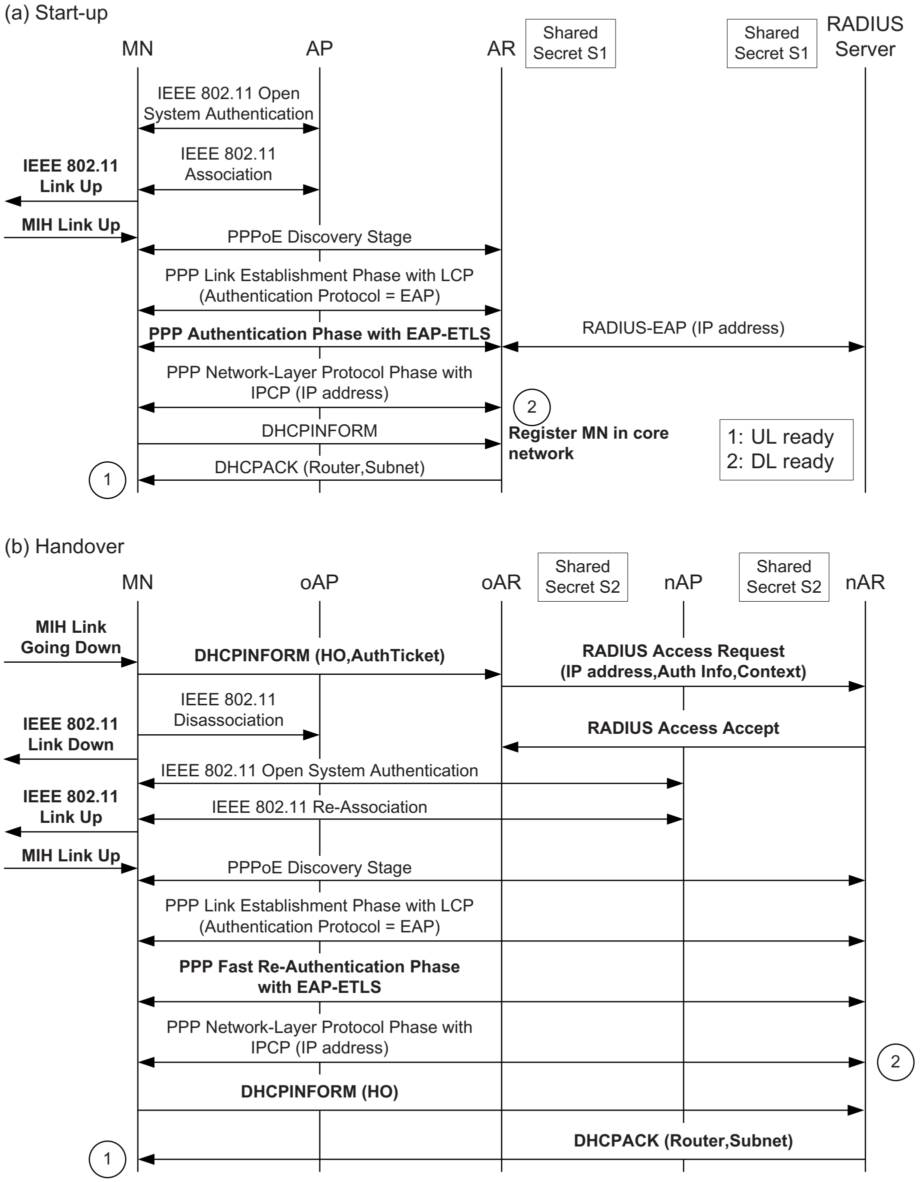

(a) Start-up
MN
AP
AR
Shared Secret S1
Shared Secret S1
RADIUS Server
IEEE 802.11 Open System Authentication
IEEE 802.11 Association
IEEE 802.11 Link Up
MIH Link Up
PPPoE Discovery Stage
PPP Link Establishment Phase with LCP (Authentication Protocol = EAP)
PPP Authentication Phase with EAP-ETLS
RADIUS-EAP (IP address)
PPP Network-Layer Protocol Phase with IPCP (IP address)
2
DHCPINFORM
Register MN in core network
1: UL ready
2: DL ready
DHCPACK (Router,Subnet)
1
(b) Handover
MN
oAP
oAR
Shared Secret S2
nAP
Shared Secret S2
nAR
MIH Link Going Down
DHCPINFORM (HO,AuthTicket)
RADIUS Access Request (IP address,Auth Info,Context)
IEEE 802.11 Disassociation
IEEE 802.11 Link Down
RADIUS Access Accept
IEEE 802.11 Open System Authentication
IEEE 802.11 Link Up
IEEE 802.11 Re-Association
MIH Link Up
PPPoE Discovery Stage
PPP Link Establishment Phase with LCP (Authentication Protocol = EAP)
PPP Fast Re-Authentication Phase with EAP-ETLS
PPP Network-Layer Protocol Phase with IPCP (IP address)
2
DHCPINFORM (HO)
DHCPACK (Router,Subnet)
1

Fig. 1. Start-up and handover for IEEE-802.11-based access networks using PPP (extensions/adaptations in bold)

communicate with the RADIUS server located in the core network, which might be undesirable if the access and core networks are administered by different providers, we propose to use the Point to Point Protocol (PPP) and PPP over Ethernet (PPPoE) so that the actual authentication can be done between the mobile device and the Access Router (AR) in the core network. In that case, Open System Authentication is used as IEEE 802.11 authentication.

The MIH Link Up event can be used as trigger to establish a PPP connection between mobile device and AR. In the PPP Link Establishment phase based on the Link Control Protocol (LCP), the AR makes use of the Authentication-Protocol Configuration Option set to Extensible Authentication Protocol (EAP).

In the subsequent Authentication phase that is initiated by the AR as EAP Authenticator in pass-through mode, the mobile device authenticates to a back-end RADIUS server. We propose to use Transport Layer Security (TLS) and an extension to EAP-TLS (EAP-ETLS) to support Fast Re-Authentication for handovers. EAP-ETLS is designed to be downwards compatible to EAP-TLS. In EAP-ETLS, the type-data field of the EAP-Request/Identity contains the public key $Publ_{AR}$ of the AR. For the start-up procedure, $Publ_{AR}$ is not needed and can be ignored. At the end of the EAP exchange, the mobile device (also called Mobile Node (MN)) and the AR share a master key from which a session key (S_{MN-AR}) can be derived. The session key is used for securing control plane and optionally also user plane messages between MN and AR.

If the authentication is successful, the MN is allowed to proceed to PPP Network-Layer Protocol phase where it uses IP Control Protocol (IPCP) to obtain an IP address. The IP address has been handed to the AR by the RADIUS server during the PPP Authentication phase. Finally, the MN needs to obtain further L3 configuration information that may be obtained through DHCP. We propose to exploit the DHCPINFORM message as trigger to register the MN in the MPLS-based core network.

4 Handover

In order to reduce the outage period for the mobile device, the L3 handover is triggered before the L2 handover. In principle, two issues have to be solved. One is to redirect the data packets to the new L3 PoA as fast as possible, which has been solved by [5]. If data packets are redirected to the new L3 PoA and arrive there before the mobile device attaches with it, the new L3 PoA may buffer these data packets. However, it may not start delivering the buffered data packets as soon as the mobile device attaches with it since the mobile device has to be authenticated before. If authentication is done with a distant AAA server, the outage period is certainly not reduced. Therefore, the other issue is to re-authenticate the mobile device at the new L3 PoA as fast as possible, which is shown in the following.

We propose a (new) Handover (HO) Option for the DHCPINFORM message. The HO Option shall contain the IP address of the previous AR and, if broadcast by the previous AP or AR or obtained by some other means, the IP address of the new AR, otherwise the Basic Service Set ID (BSSID) of the new AP. It shall further contain an authentication ticket to provide fast re-authentication with the new AR. The authentication ticket (cf. (1)) consists of a handover sequence number (HOSeqNo), a new master key nMK (a random number chosen by the

MN), and a unique identifier MNID of the MN (e.g. L2 address, L3 address, etc.). The new master key nMK and the MNID are encrypted with a randomly chosen secret key SK so that the receiving old AR cannot decrypt them. The handover sequence number is used to prevent replay attacks and to avoid misconfigurations in the core network. The authentication ticket is itself protected with the secret key S_{MN-oAR} (cf. Section 3 where it is denoted as S_{MN-AR}) used between MN and old AR.

$$AuthTicket := \{HOSeqNo, (nMK, \mathrm{MNID})_{SK}\}_{S_{MN-oAR}} \tag{1}$$

When IEEE 802.21 is used, the Link Going Down event issued at the MN on L2 can be used as trigger for the DHCPINFORM message.

On receipt of the DHCPINFORM message (cf. Fig. 1 (b)), the old AR evaluates the HO Option enabling it to contact the new AR for context transfer. We propose to exploit RADIUS that is subject to continuous extensions for different purposes. The RADIUS message shall contain $(nMK, \mathrm{MNID})_{SK}$ decrypted from (1), and the context of the MN, i.e. the MNID and all further information that is necessary to receive and send IP packets from and to the MN at the new AR.

Note that the context transfer can be performed in parallel to the L2 handover, which is done by disassociating from the previous AP and re-associating with the new AP. After PPPoE Discovery and the PPP Link Establishment phase, the PPP Authentication phase can be kept short due to the context transfer. This phase is thus called Fast PPP Re-Authentication phase.

On receipt of the EAP-Request/Identity containing the public key $Publ_{nAR}$ of the new AR, the MN sends an EAP-Response/Identity to the new AR where the type-data field contains both the $AuthTicket$ and the following $ReAuthTicket$.

$$ReAuthTicket := \{\mathrm{MNID}, oAR, (SK)_{Publ_{nAR}}, HMAC_{nMK}(Publ_{nAR})\} \tag{2}$$

$HMAC_{nMK}(Publ_{nAR})$ is a Hashed Message Authentication Code (HMAC) computed over $Publ_{nAR}$ and seeded with nMK, the same random number that the MN used in the authentication ticket sent to the old AR (cf. (1)). On receipt of $ReAuthTicket$, the new AR uses the MNID to retrieve $(nMK, \mathrm{MNID})_{SK}$. If the new AR has not yet received the RADIUS message containing $(nMK, \mathrm{MNID})_{SK}$, it sends a RADIUS message including $AuthTicket$ to the previous AR in order to trigger the fast re-authentication. Once in possession of $(nMK, \mathrm{MNID})_{SK}$, it decrypts SK from (2) with its private key and is then able to decrypt nMK and MNID. The decrypted MNID serves to verify that the context the new AR received from the old AR indeed belongs to the MN having sent the $AuthTicket$. The HMAC serves to verify that $ReAuthTicket$ has been sent by the same MN as $AuthTicket$. As the new AR receives the authentication ticket from a trustworthy partner, i.e. from the old AR, over a secure channel, the MN is authenticated and the new master key nMK is established. With only three messages and without the necessity of contacting a RADIUS server the procedure is fast and, with the exception of two public-key operations, the computational overhead is low.

In the subsequent PPP Network-Layer Protocol phase, the MN asks to be assigned the same IP address as before. In order to notify the new AR on its arrival, the MN sends a DHCPINFORM containing the HO Option, yet without *AuthTicket* that is not necessary there. If the new AR has not yet received a handover notification message from the previous AR, it sends a handover indication message to the previous AR in order to trigger the handover procedure. The new AR finally acknowledges the DHCPINFORM with a DHCPACK containing all necessary configuration information such as gateway address and subnet mask. The DHCPACK completes the handover.

5 Conclusion

This paper has shown that MPLS-based micro-mobility solutions may be triggered before the corresponding L2 handover by exploiting IEEE 802.21. The outage period is, however, only reduced for the mobile device when both packet redirection at the previous L3 PoA and the re-authentication at the new L3 PoA are done in a fast and efficient way. Both issues can be solved as shown in this paper. DHCP is exploited for handover indication. A new HO Option is introduced in order to distinguish a handover trigger from a conventional DHCP message. Security has been addressed by EAP-TLS. In order to allow for fast re-authentication during handover, EAP-ETLS has been introduced as extension. Inter-technology handovers have not been covered in this paper. For inter-technology handovers, a mobile device has to be equipped with at least two L2 interfaces of different technologies. Furthermore, a L2 interface change requires a mobility management entity in the mobile device. The issue of inter-technology handovers thus remains for further study.

References

1. Ren et al., Z.: Integration of mobile ip and multi-protocol label switching. In: IEEE International Conference on Communications. (2001)
2. Vassiliou et al., V.: A radio access network for next generation wireless networks based on multi-protocol label switching and hierarchical mobile ip. In: Proceedings of the 56th IEEE Vehicular Technology Conference. (2002)
3. Vassiliou et al., V.: M-mpls: Micromobility-enabled multiprotocol label switching. In: IEEE International Conference on Communications. (2003)
4. Chiussi et al., F.: A network architecture for mpls-based micro-mobility. In: IEEE Wireless Communications and Networking Conference. (2002)
5. Persaud et al., R.: An mpls-based handover solution for cellular networks. In: 19th International Teletraffic Congress (ITC). (2005)

A Comparative Performance Study of IPv6 Transitioning Mechanisms - NAT-PT vs. TRT vs. DSTM

Michael Mackay and Christopher Edwards

Computing Department, InfoLab 21, Lancaster University,
Lancaster, LA1 4WA, UK
{m.mackay, ce}@comp.lancs.ac.uk

Abstract. One of the major challenges faced by the IPv6 community in recent years has been to define the scenarios in which transitioning mechanisms should be used and which ones should be selected given a specific scenario. This paper aims to supplement this by presenting the results of a comparative evaluation carried out on three major IPv6 interoperation mechanisms; NAT-PT, TRT and DSTM. This work attempts not only to determine the outright performance of each mechanism against the other but also against a theoretical evaluation of the specification. Our results show that while DSTM performs well both NAT-PT and TRT place significant overheads on the network.

Keywords: IPv6, Transitioning Mechanisms, NAT-PT, TRT, DSTM.

1 Introduction

One of the major challenges faced by the IPv6 community in recent years has been to define the scenarios in which transitioning mechanisms should be used and which ones should be selected given a specific scenario. This is well illustrated by the IETF V6OPS working groups [1] who have led this process by defining and analyzing four broad IPv6 deployment scenarios; Unmanaged [2], Enterprise [3], ISP [4] and 3GPP [5], which each represent a key area for IPv6 deployment. As such, the thorough completion of this process is critical since its outcome may largely determine the future use of all such mechanisms.

Transitioning mechanisms can generally be divided into three groups according to their operation and functionality: Tunnelling, Translation and Dual Stack. We choose however to focus on mechanisms that support the interoperation between IPv4 and IPv6 which includes both translator and dual stack mechanisms. This paper presents the results of a comparative evaluation carried out on three interoperation mechanisms; NAT-PT, TRT and DSTM which each allow IPv4 and IPv6 hosts to communicate. Our aim is to supplement the ongoing analysis work with a comparative evaluation to show how each performs under test conditions. This paper does not attempt to evaluate the implementation of each mechanism but rather concentrates on extracting the mechanism-specific properties to test each *transitioning approach* against the other.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1125 – 1131, 2006.

Hereafter this paper is organised as follows, section 2 conducts a theoretical performance analysis in an attempt to extract any inherent qualities. Section 3 presents the testing and section 4 concludes with an analysis of our results.

2 Theoretical Mechanism Evaluation

This section presents a theoretical evaluation of each mechanism to estimate the test performance we can expect in each case. In each case, diagrams outline the tasks that must be performed with the darker shading indicating the more complex operations.

2.1 NAT-PT Performance Evaluation

NAT-PT (Network Address Translation - Protocol Translation) [6] extends NAT to provide a translator that binds IPv6 addresses to IPv4 addresses from a local pool and keeps state on sessions passing through it. One weakness of NAT-PT is its inability to translate upper layer protocols (e.g. DNS) using embedded IP addresses requiring the use of application level gateways (ALGs). While NAT-PT is likely to be deployed to some degree, it is now unpopular with the majority of the IPv6 community due to it over-complex approach and has recently been moved to experimental standard.

Session Initiation. In NAT-PT this will incur significant overheads due to the address allocation and the state that is kept on each session which must be setup during initialisation. Fig 1 shows the steps to initialise a session in NAT-PT with the heavyweight aspects including the address allocation translation of the first packet.

Operation. Once the session is in progress, translation is done on a per-packet basis with lookups needed to retrieve the address bindings. During translation, IP headers are completed first before upper layer (TCP/UDP) protocols. Finally any higher-level protocols (e.g. FTP) must be translated before the packet is forwarded. This process is shown in Fig 2. As such, the overhead introduced will vary according to the packet

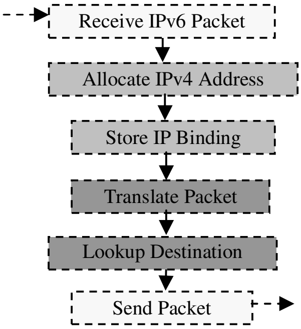


Fig. 1. Initiation tasks for NAT-PT

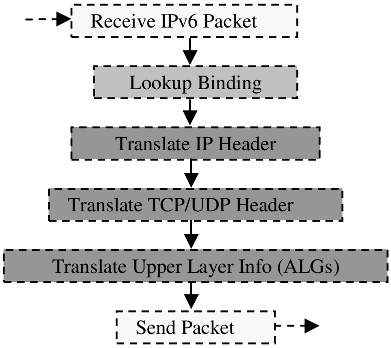


Fig. 2. Per-Packet translation for NAT-PT

being translated and depending on the complexity of the packet, these overheads may be quite significant. Overall, we expect NAT-PT to perform quite poorly, session initiation will be significant while bi-directional per-packet translation suggest that operational performance will be poor also.

2.2 TRT Performance Evaluation

TRT (Transport Relay Translator) [7] transparently relays TCP/UDP connections between IPv4 and IPv6 and between the source and destination. As with NAT-PT, it keeps state on sessions and cannot handle embedded IP addresses. As a relay, TRT is reasonably efficient and is now the preferred translation-based solution. We expect therefore that TRT will perform better than NAT-PT but still introduce significant overheads as packets are translated at the transport layer before being forwarded.

Session Initiation. On initiation TRT must setup two TCP/UDP connections, from the IPv6 host to the relay and from the relay to the IPv4 host necessitating a certain amount of state being configured. Fig 3 gives an overview of TRT initialisation which is the simplest and therefore (we expect) the quickest on test.

Operation. Once the initialisation in complete, a limited amount of processing in necessary as flows are relayed between connections as shown in Fig 4. The only real overheads introduced are a lookup to establish the outgoing address and the construction and sending of the packet. Upper layer protocols such as FTP must again be handled via an ALG. The most significant aspect of normal TRT operation will be in the relaying of packets between connections. This necessitates the packet traversing up one IP stack to the transport layer and back down the other, however, we expect TRT to perform better than NAT-PT in most aspects.

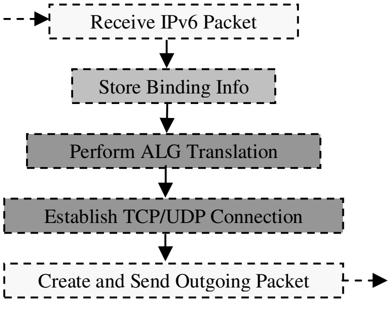


Fig. 3. Initiation of TRT

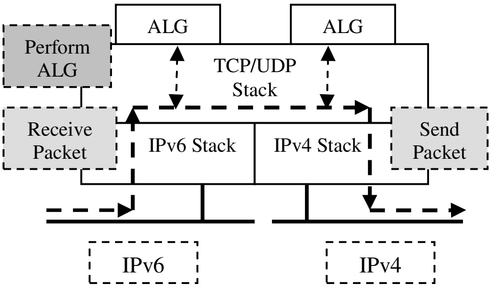


Fig. 4. Operation of the TRT mechanism

2.3 DSTM Performance Evaluation

DSTM (Dual Stack Transition Mechanism) [8] uses automatic tunnelling to enable Dual Stack enabled hosts in an IPv6-only network to acquire a temporary IPv4 address and communicate with IPv4 hosts. It is composed of a Server for address allocation, a Tunnel End Point (TEP) and the hosts.

Session Initiation. The DSTM initiation process is complex, involving communication between all three components. On initialisation, a DSTM client in the IPv6 host will contact the Server which replies with both an address allocation and the address of the TEP. The host then encapsulates the first IPv4 packet and sends it to the TEP where the packet is decapsulated, the IPv4<->IPv6 binding is stored and the IPv4 packet is sent. This process is shown in Fig 5 indicating which components are involved in each step. The overheads in this process will be incurred during the communication between the components prior to traffic flow starting. **Operation -** Once the session is in progress, DSTM is far more straight-forward as its operation simply involves an IPv4-over-IPv6 tunnel with the IPv6 host and TEP performing (d)encapsulation on packets sent. The TEP must also do a lookup on each IPv4 packet received to determine the destination IPv6 address. Fig 6 shows this from the perspective of a returning IPv4 packet. Once DSTM is established, its overheads will be minimal as only simple (d)encapsulation and forwarding is necessary. As such, we expect initiation performance to be poor but the operational should be excellent.

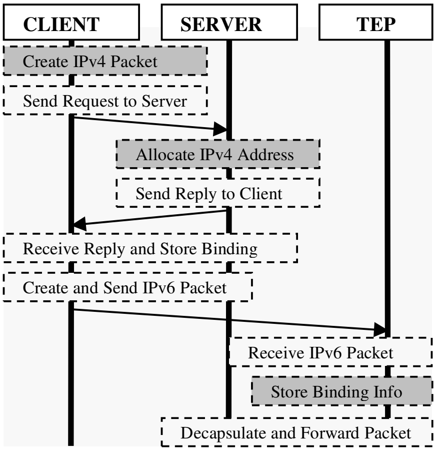


Fig. 5. Initiation of the DSTM mechanism

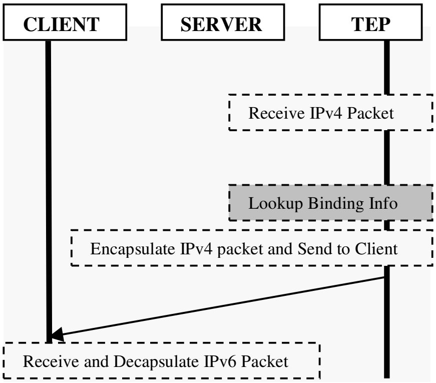

CLIENT
SERVER
TEP
Receive IPv4 Packet
Lookup Binding Info
Encapsulate IPv4 packet and Send to Client
Receive and Decapsulate IPv6 Packet

Fig. 6. Reverse traversal of DSTM

3 Results

The aim of our evaluation was to test the mechanisms over a common 100 Mbps test network to give a better indication of the relative performance of each approach. The test network comprised of a small IPv6-only subnet behind a Dual Stack gateway with hosts on either side to locate the testing tools. For our tests, the ETRI implementation of NAT-PT [9], pTRTd from Litech Systems [10] and ENST DSTM [11] over Linux Red Hat 9.0 were used with each result representing the average performance from a number of tests. The testing was done using IPERF [12] to benchmark mechanism performance and MGEN [13] to generate network traffic flows.

The testing comprised of three stages with the initial testing establishing the optimum performance of each mechanism to provide a direct comparison of each including initiation performance (from receipt of the first packet to it being sent on the external interface). The next phase tested performance under increasing levels of simplex (IPv6 to IPv4) traffic with the final phase testing duplex traffic performance to represent realistic network conditions. To establish the loading increments for each testing phase, IPv4-only testing was done first and a reasonable scale selected.

3.1 Initial Benchmarking Results

The results of the initial performance testing are shown in Fig 7 with the initiation testing results shown in Table 1. These show to good effect the relative performance of each mechanism in comparison to IPv4. DSTM is the best-performing mechanism, averaging at about 90 Mbps with TRT showing 40 Mbps and NAT-PT only slightly worse at 32 Mbps. This is what we would expect to see in a direct comparison and shows the performance advantage DSTM has over translators giving results only slightly inferior to IPv4-only. The initiation tests again reinforce what we expected to see with TRT clearly the best averaging around 0.26 milliseconds followed by NAT-PT at 0.81 milliseconds with DSTM the slowest at over 1.35 milliseconds on average.

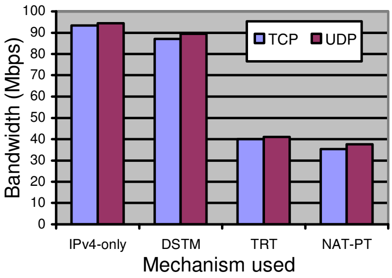


Fig. 7. Optimum mechanism testing results

Table 1. Mechanism Initialisation Results

Device	Min. (ms)	Max. (ms)	Av. (ms)
TRT	0.261	0.269	0.264
NAT-PT	0.751	0.899	0.816
DSTM	1.317	1.422	1.353

3.2 Simplex Testing Results

The simplex test results are shown in Fig 8 and highlight the performance against a gradually increasing IPv6 -> IPv4 traffic flow. The **IPv4-only** test showed that the performance decreases roughly in increments of 10Mbps per 1000 packets per second (pps) of loading introduced. One interesting result we noticed was that once the loading increases past a certain point, (around 5000pps) the bandwidth curve tended to level out with no further performance degradation experienced. The **NAT-PT** results were poor in comparison to both the IPv4-only results and the other mechanisms tested. The performance results show it performed consistently in the range on 30Mbps but that the traffic load had a much less pronounced affect on mechanism performance. The **TRT** results again show it to be quite resilient to network load with performance consistently in the 30–40Mbps range and a slight decline in performance as the load increases. It was consistently worse than IPv4 but in all cases performance was superior to NAT-PT. The **DSTM** results clearly show it to be the best performing mechanism tested. In an unloaded network it performed similar to IPv4, around the 90Mbps mark, also falling in a similar way under load.

3.3 Duplex Testing Results

The duplex testing results as shown in Fig 9 show how mechanism performance was affected by both IPv6 -> IPv4 and IPv4 -> IPv6 traffic. The **IPv4-only** results show that performance suffers heavily in this scenario. The bandwidth rapidly falls until a rate of 50 pps where it levels out and falls gradually reaching 9Mbps at 100 flows per host. **NAT-PT** also performed badly in the duplex tests managing results only up to 30 flows per host. Our results show that performance fell sharply from 34 to 18Mbps in the first test and thereafter slowly degraded until it failed testing 40 flows per host. Unfortunately, no accurate test results could be gathered for **TRT** because the approach dictates that it be IPv6-initiated without the use of a DNS-ALG meaning IPv4 initiated traffic is not possible in this case. The **DSTM** results again show it to be the best-performing mechanism tested. It again performs in a similar manner to IPv4-only, initially dropping rapidly before levelling off. As with NAT-PT, DSTM failed to register a complete set of results and failed while testing the 70 pps scenario.

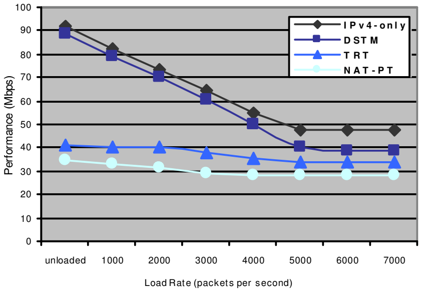


Fig. 8. Combined simplex performance

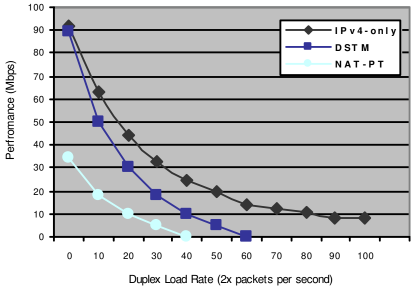


Fig. 9. Combined duplex performance

4 Conclusions

The results of the testing both reinforced what we expected to see and produced some interesting results that highlight the behaviour of these mechanisms. The results generally give the order IPv4-only, DSTM, TRT and NAT-PT which is essentially what we predicted. NAT-PT performance was quite poor, TRT outperformed NAT-PT and DSTM was very impressive in its proximity to IPv4 performance. Based on these results there is little to recommend about NAT-PT, also given its move to ‘experimental’ it is the least preferable solution considered here. TRT fared better and while it is not on a par with DSTM it represents the best translator device. DSTM however is the ‘fastest’ mechanism evaluated but is the most complex to deploy and IPv4 address resources must be committed to make it scalable. Further work will include simulations to test mechanism scalability in larger networks and testing of other implementations (possibly *BSD) to negate any implementation-specific anomalies.

References

1. IETF v6ops Homepage, http://www.ietf.org/html.charters/v6ops-charter.html.
2. C. Huitema, R. Austein, S. Satapati, R. van der Pol, “Evaluation of IPv6 Transition Mechanisms for Unmanaged Networks”, RFC 3904, September 2004.
3. J. Bound, Y. Pouffary, T. Chown, D. Green, S. Klynsma, “ IPv6 Enterprise Network Analysis”, draft-ietf-v6ops-ent-analysis-04.txt, January 2006, work in progress
4. M. Lind, V. Ksinant, S. Park, A. Baudot, P. Savola, “Scenarios and Analysis for Introducing IPv6 into ISP Networks”, RFC 4029, March 2005.
5. J. Wiljakka (ed.), “Analysis on IPv6 Transition in Third Generation Partnership Project (3GPP) Networks”, RFC 4215, October 2005.
6. G. Tsirtsis, P. Srisuresh, “Network Address Translation - Protocol Translation (NAT-PT)”, RFC 2766, February 2000.
7. J. Hagino, K. Yamamoto, “An IPv6-to-IPv4 Transport Relay Translator”, RFC 3142, June 2001.
8. J. Bound (Ed.), "Dual Stack IPv6 Dominant Transition Mechanism (DSTM)”, draft-bound-dstm-exp-04.txt, October 2005, work in progress.
9. NAT-PT implementation, http://www.ipv6.or.kr/english/natpt-overview.htm.
10. TRT implementation, http://v6web.litech.org/ptrtd/.
11. DSTM implementation, http://www.ipv6.rennes.enst-bretagne.fr/dstm/.
12. The Iperf Toolset Homepage, http://dast.nlanr.net/Projects/Iperf/.
13. The Mgen Toolset Homepage, http://mgen.pf.itd.nrl.navy.mil/.

CAC: Context Adaptive Clustering for Efficient Data Aggregation in Wireless Sensor Networks

Guang-yao Jin and Myong-Soon Park*

Dept. of Computer Science and Engineering, Korea University,
Seoul 136-701, Korea
{king, myongsp}@ilab.korea.ac.kr

Abstract. Wireless sensor networks are characterized by the widely distributed sensor nodes which transmit sensed data to the base station cooperatively. However, due to the spatial correlation between sensor observations, it is not necessary for every node to transmit its data. There are already some papers on how to do clustering and data aggregation in-network, however, no one considers about the data distribution with respect to the environment. In this paper a context adaptive clustering mechanism is proposed, which tries to form clusters of sensors with similar output data within the bound of a given tolerance parameter. With similar data inside a cluster, it is possible for the cluster header to use a simple technique for data aggregation without introducing large errors, thus can reduce energy consumption and prolong the sensor lifetime. The algorithm proposed is very simple, transparent, localized and does not need any central authority to monitor or supervise it.

1 Introduction

In the case when a sensor network is sensing simple data, such as the temperature of a room exposed to sunlight, it can be assumed that there will be several regions where measured temperature is similar under a specific tolerance. As the sun moves from East to West, those areas are going to change slowly as well. The problem is that certain regions would be in the adjacent area of different clusters, thus those adjacent cluster headers would have to send some overlapped data to the base station for correct data aggregation. This will generate more network traffic and energy consumption.

For example, in Figure 1, there are two clusters and two different temperature regions. Each cluster header only needs to transmit one data to the base station after data aggregation. After a certain time period, the temperature distribution will change as shown in Figure 2. In this case, existing approaches of data aggregation (e.g., work out an average as proposed in [5]) would produce two representative temperatures per region (e.g., a1 and a2 in region a) in order to maintain the high data correlation in a cluster which has a localized property. Thus, each cluster header will transmit two data representatively and together transmit four. However, some sub-regions such as (a2) and (b1) have the same localized property and produce identical representative data correspondingly. It is unnecessary to consume energy to send the same data item twice (e.g., a2 and b1).

* Corresponding author.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1132 – 1137, 2006.

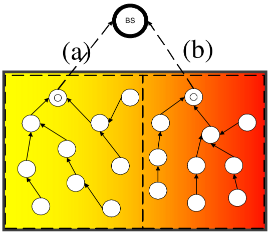


Fig. 1. Initial temperature Distribution

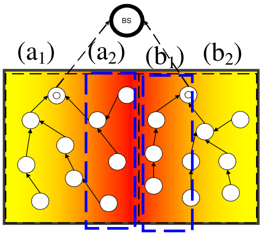


Fig. 2. The temperature distribution was changed

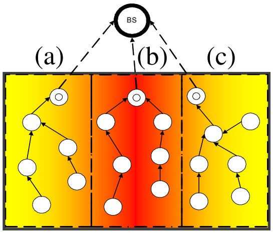


Fig. 3. After adaptive clustering was performed

Our approach is to use an adaptive re-clustering algorithm in order to faithfully represent the physical reality. As shown in Figure 3, since the temperature distribution (i.e., context) changed, the two regions (or clusters) adaptively change into three regions (clusters) without a centralized (global) component processing, and each region send a representative data separately, thus there are all together three data need to be send. In this way, our approach could produce correct representative data (better represents the physical reality), reduce energy consumption of sensor nodes and prolong sensor network life.

This paper is organized as follows. In section 2, the related work is presented. Section 3 proposes our algorithm, and Section 4 discusses the simulation results of our algorithm. Section 5 concludes the paper with future work.

2 Related Work

There have been some researches on clustering in wireless sensor networks as discussed in [1], [2], [3], [6], [7]. In [1] the cluster heads are identified once during network deployment by a central controller. Also in [2] the clusters are formed during the actual physical deployment of the sensor networks which have to be planned by the network designers in advance. However, our proposed algorithm does not need a central controller and clustering is performed dynamically through the sensor network lifetime. [3] and [4] require the priori information of location and the initial sensor energy. However, in our approach such information is not needed to form and update the clusters dynamically. The LEACH protocol [6] for clustering and cluster-head determination was proposed, which goal is to organize clusters based on the energy level of sensor nodes and to re-circulate the elected cluster-header inside a cluster in order to save battery power of the nodes. However, the algorithm this paper proposed is concerned about organizing clusters based on the data they sense, i.e. geographically partitioning physical space into clusters of correlated data, and thus making data aggregation more effectively and faithfully represent the physical reality. An improved version of LEACH, which is called LEACH-C [7], has a set-up phase for initial cluster-head computation by the base station. However, our approach does not require such initial step, since it concerns with forming the clusters based on their data output and the re-clustering is done by the network itself in a certain region.

In summary, the main feature of our algorithm compared with existing clustering algorithms is a very effective technique in sensing which is localized and does not require computations by some higher (central) entities and re-clustering is performed dynamically to keep high data correlation.

3 Context Adaptive Clustering (CAC)

As discussed in Section 1, data aggregation can be performed efficiently and correctly when data from different sensors are highly correlated. But if the collected data change over time due to the change of actual physical environment, data correlation must be changed correspondingly. One of the CAC goals is to maintain the high data correlation within a cluster and therefore save more sensor energy.

3.1 Assumption

In this paper we assume that the sensed data is changing smoothly over a long time period, and the data has a regional property, i.e. in one specific area data is similar, so cluster can be formed and data can be aggregated by the cluster header.

3.2 The Proposed Algorithm

At initial deployment, how many geographic clusters a sensor network region can be partitioned is manually determined to form initial clusters. Adaptive re-clustering is performed locally by header nodes using the proposed approach recursively until stability is achieved. The stability is defined as a state that re-clustering is ended in the network and all clusters have correlated data, based on the threshold tolerance.

More formally, a set of the correlated nodes in a cluster, e.g. $\sigma(d,\delta)$, is defined to determine whether a node d_i belongs to the cluster or not. Its input parameters are the aggregated data value (d) and the tolerance parameter (δ). If a node d_i is in $\sigma(d,\delta)$ (i.e., $d_i \in \sigma(d,\delta), i = 1,2...n$, where the cluster size is n), node d_i belongs to the cluster. That is, a node belongs to the cluster if and only if its output data is equal to the aggregated value (d) or bounded around it given the tolerance parameter δ . When the time passes by, the data distribution changes smoothly. If there are p nodes whose data do not belong to $\sigma(d,\delta)$ (i.e., $d'_j \notin \sigma(d,\delta), j = 1,2...p,$ where $p < n$), data of these nodes are separated into m ($1 \le m \le p$) different ranges. Then the cluster header needs to use $m+1$ data to represent the whole data correctly assuming a simple algorithm, such as computing an average, for data aggregation. $D = \{d_j \mid d_j \notin \sigma(d,\delta), j = 1,2...p\}$ is used to indicate the p nodes, and $D'_k, k = 1,2,...,m$ indicates the m different ranges, here $D = \sum_1^m D'_k$. The worst case takes place when $(m = p) \& (p = n-1)$. In this case there are no benefits from data aggregation.

Given a threshold M , when m becomes bigger than M , the cluster header initially generates the new headers list $H = \{h_k \mid h_k \in D'_k, k = 1,2,..., m\}$ which contains new possible cluster header candidates and then the re-clustering will be performed.

```
Loop
  Gathering data from $d_1$ to $d_n$ ;
  Compute $D$ ;
  Compute $m$ and $D'_k$ ;
If $m \geq M$ then
  Generate $H$ ;
  Broadcast re-clustering ( $\sigma(d,\delta), H$ );
Else
  Do data aggregation
```

Fig. 4. Algorithm to decide when to start re-clustering

```
Read its data as $d'$ ;
If $my\_id \in H$ then
  // consider itself as header and broadcast
  Broadcast re-clustering ( $\sigma(d',\delta)$, Null);
Else
  If $d' \in \sigma(d,\delta)$ then
    Join the new cluster;
    Forward re-clustering ( $\sigma(d,\delta), H$ );
```

Fig. 5. Algorithm to decide to join the cluster

Algorithm in Figure 4 is executed by the current cluster header to decide whether to initiate re-clustering or not. If re-clustering is needed, cluster header forms a list of nodes that fall out of its current range and broadcasts the list to the nearby nodes. Nearby nodes examine their current sensor output and decide whether they join one of the new possible clusters or stay in their current one as explained in Figure 5.

A node receiving the re-clustering command either considers itself as a cluster header and re-broadcasts the command, or joins a new cluster based on their current sensor output and forwards the re-clustering command to other nodes in its vicinity. If a node is neither a new cluster header nor interested in joining a new cluster, it would disregard the received command, which would stop and bound the algorithm to a certain geographic region.

4 Experimental Results

To validate the energy efficiency by reducing data items that have to be transmitted to satisfy the tolerance parameter, we have simulated both the LEACH and our proposed mechanism in NS2. In our experiments, we used a 100-node network where nodes were randomly distributed between (0, 0) and (100, 100). The radio model adopted in this experiment is based on [8]. The function used for data aggregation was computing an average of the data received from the nodes in a cluster, which was computed based on the nodes' location in the area and the current location of the data source

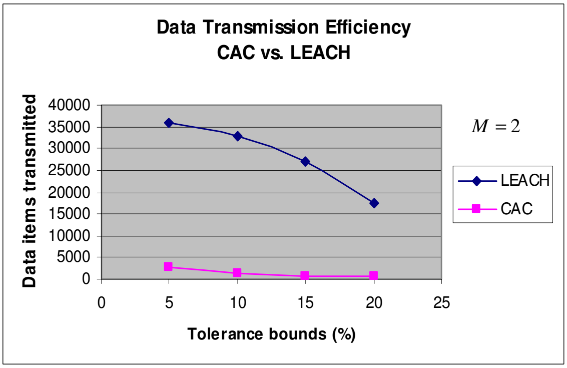


Fig. 6. Simulation results for 100 nodes

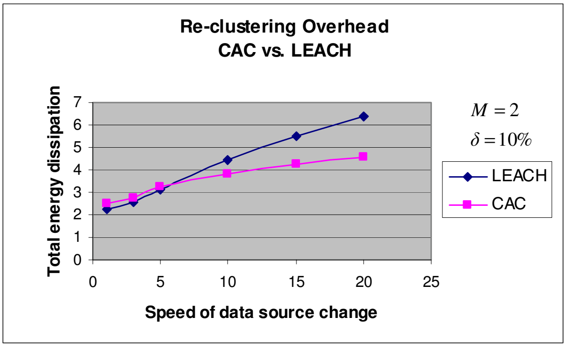


Fig. 7. Total energy dissipation of CAC vs. LEACH

using the Euclidian distance formula for the 2D-plane, considering the fact that the node which is the nearest to the data source would have a maximum output of 100 and the farthest one would have a minimum output of 0, and the outputs of other distributed sensor nodes are evenly spaced between 0 and 100 accordingly.

For each time increment, data is being sent from all the clusters. A cluster header would send the aggregated data item and may still send a data item from the sensors that do not fit in the tolerance criterion, but however, in this case, re-clustering would be performed. Results of the simulation are given in Figure 6 for the case of 100 sensor nodes. Our approach in Figure 6 shows, in terms of how many data items are to be transmitted, a 31.1-fold improvement for 10% tolerance and a 29.78-fold improvement for 20% tolerance compared with LEACH. Thus, by having fewer transmissions, implicitly power consumption is reduced without affecting reliability or availability.

In case of the overhead imposed by CAC, according to our assumptions of slowly changing data, re-clustering will not be performed very frequently, so the overhead can be considered negligible. In case the data changes very rapidly, such an overhead for re-clustering would be significant, thus other solutions than ours would be more suitable. Results of such experiment are given in Figure 7.

For smooth changing data, CAC reduces the number of data items that have to be transmitted and thus enhance the network lifetime by requiring less data transmissions.

5 Conclusions and Future Work

In this paper, an energy-efficient algorithm is proposed that can generate clusters in a sensor network for data aggregation and adapt the clusters by performing re-clustering depending on the data changes caused by the environment changes. Resulting clusters have similar data that can be easily aggregated by the cluster header without introducing large error in the aggregated data output using an efficient, computable and inexpensive algorithm, and thus power consumption is reduced. Further more, CAC is localized, that is, a re-clustering is performed regionally and independently from both other clusters in some other area and the central authority, such as base station.

We would like to focus our future work on how to decide the optimal M for CAC, since the parameter M directly affects the performance of CAC.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korea Government (MOEHRD) (KRF-2005-211-D00274).

References

1. Jason Tillet, Raghuveer Rao and Ferat Sahin, “Cluster-Head identification in ad hoc sensor networks using particle swarm optimization”. Proc. of the IEEE International. Conference on Personal Wireless Communication, 2002.2.
2. Wei-Peng Chen, Jennifer C. Hou and Lui Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks“. Proc. 11th IEEE Conf. on Network protocols, 2003.
3. Kostuv Dasgupta, Konstantinos Kalpakis and Parag Namjoshi, “An efficient clustering-based heuristic for data gathering and aggregation in sensor networks”. Proc. of the IEEE Wireless Communications and Networking Conference, March 16-20, 2003.
4. Konstantinos Kalpakis, Koustuv Dasgupta, and Parag Namjoshi, “Maximum lifetime data gathering and aggregation in wireless sensor networks". Proc. of the IEEE International Conference on Networking (ICN'02), Atlanta, Georgia, August 26-29, 2002. pp. 685-696.
5. J. Considine, F. Li, G. Kollios, and J. Byers, “Approximate Aggregation Techniques for Sensor Databases”. Proc. of the 20th International Conference on Data Engineering, 2004
6. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication protocol for wireless microsensor networks”. Proc. of the 33rd Hawaii International Conference on System Sciences, 2000.
7. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks”. IEEE Transactions on Wireless Communications, vol.1, no.4, October 2002.
8. Jain-Shing Liu; Lin, C.-H.P. “Power-Efficiency Clustering Method with Power-Limit Constraint for Sensor Networks”. Proc. of the 2003 IEEE International, 9-11 April 2003.

On the Performance of Cooperative Diversity in Infrastructure-Based Networks with Two Relays

Jun Yeop Jung

Yonsei University, Dept. of Electrical and Electronic Engineering,
134 Shinchon-Dong, Seodaemun-Gu, Seoul 120-749, Korea
amigos97@yonsei.ac.kr
http://mcl.yonsei.ac.kr

Abstract. In this paper, the performance of four cooperative relaying schemes, which are classified by requiring a quantity of feedback information, is evaluated in the high SNR region and is compared with that of MIMO relaying schemes with two antennas. For amplify-and-forward (AF) relay, all schemes except the second hop selection scheme provide the second order diversity gain in high SNR region. For decode-and-forward (DF) relay, despite using more channel information, the coherent BF scheme does not offer the second order diversity. A system level simulation is evaluated to analyze the effects of user distribution in terms of the average capacity. For AF relay, the capacity of the cooperative relaying model is higher than that of the MIMO relaying model as about 0.2 bps/Hz. For DF relay, the MIMO relaying model provides capacity gain about larger than 0.3 bps/Hz for the coherent beamforming and the Alamouti-based scheme since the effect of array gain at relay for these schemes is more dominant factor to increase the capacity compared with the effect of reducing path loss.

1 Introduction

Recently there has been increasing interest in the concept of augmenting the infrastructure-based networks with relay in order to provide high data rate and expand service coverage in a cost efficient manner[1,2]. Moreover, the present paper evaluates the performance of cooperative diversity using relay. Based on the ideas of user cooperation diversity [3], Laneman et al. [4] propose cooperative protocols for the three-terminal case, and it is shown that diversity gains can be achieved.

In this paper, the performance of four cooperative relaying schemes, which are classified by requiring a quantity of feedback information, is evaluated in the high SNR region and is compared with the performance of MIMO relaying schemes with two antennas. The first, performance is measured in terms of capacity, outage probability and cooperative diversity in only short-term fading channel environment. Then, the performance is investigated in environment where long-term fading is considered for more realistic assumption.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1138–1143, 2006.

2 Channel and Relay Model

In our scenario, it is assumed that BS and mobiles can communicate via relay, user mobility is low. We assume that the channel is considered time-invariant over at least one transmission cycle (block fading). BS transmission slot separated from relay transmission slot in order to avoid interference. The discrete-time baseband equivalent model of the channel with two relays which use same frequency band is shown in Fig.1. From now on, n' is used as symbol index of the first slot, and n is used as symbol index of the second slot. For cooperative diversity, a signal during the first slot received at the relay and the received signal during the second slot at the mobile are

$$r_i[n'] = h_{br_i} s[n'] + z_{br_i}[n'] \tag{1}$$

and

$$y[n] = h_{r_1 m} x_1[n] + h_{r_2 m} x_2[n] + z_m[n] \tag{2}$$

for $n, n' = 1, ..., N$.

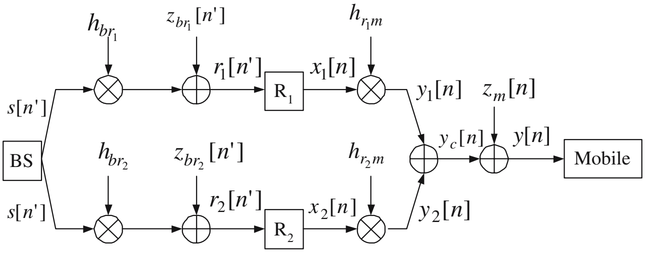


Fig. 1. Discrete-time baseband equivalent relay channel

The BS transmits signal as $s[n']$ for first slot. During this interval, the i-th relay processes received signal, $r_i[n']$ and AF relay retransmits signal

$$x_i[n] = w_i[n] g_i[n] r_i[n'] \tag{3}$$

for $n, n' = 1, ..., N$. For AF relay case, the received signal at relay is retransmitted after its power was amplified by the following amplifier gain,

$$g_i[n] = \sqrt{\frac{P_R}{P_B |h_{br_i}|^2 + N_{r_i}}} \tag{4}$$

where the amplifier gain depends upon the received power. For DF relay the transmitted signal at relay is re-encoding version of the received signal. We assume that the relay might fully decode without error. The relay transmits the signal

$$x_i[n] = w_i[n] \hat{s}[n'] \tag{5}$$

for $n, n' = 1, ..., N$. $\hat{s}[n']$ denotes re-encoded signal of received signal from BS. The total power of two relays transmitted signals is P_R for fair comparison among different transmission schemes. $w_i[n]$ denotes the weight which depends on transmission schemes. The weight constraint is given by

$$|w_1[n]|^2 + |w_2[n]|^2 = 1. \tag{6}$$

3 Outage Probability Performance

We focus on nonergodic scenarios, and evaluate performance in terms of outage probability. The channel capacity C for an instantaneous SNR is given by

$$C = \frac{1}{2} log(1 + SNR_{received}) \tag{7}$$

where $SNR_{received}$ is the received signal to noise ratio at mobile. For a target spectral efficiency R, $C < R$ denotes the outage event, and $Pr[C < R]$ denotes the outage probability which can be approximated in high-SNR region.

3.1 Cooperative Relaying Schemes Using Two Relays with One Antenna

Four cooperative relaying schemes are considered: the coherent beamforming scheme, Alamouti-based scheme, the optimal selection scheme and the second hop selection scheme. The coherent beamforming scheme uses single antenna of two relays like maximal ratio transmission [8] in MIMO. Alamouti-based scheme is proposed by an Alamouti-based multi-user space-time diversity system of [6,7]. Since the scheme in [6,7] uses two frequency bands, modified transmission scheme which use single frequency band is used for fair comparing other schemes. The optimal selection scheme selects the path of better total channel condition out of two relay paths. All CSIs of relay channel should be used for deciding optimal path. Then selected relay transmits signals with power, P_R and the other relay stop transmitting. In other words, magnitude of weight of selected relay is one and the other is zero. The second hop selection selects relay which has better channel condition between relay and mobile. It can be regard the sub-optimal scheme, which makes it work with partial information.

For AF relay, the capacity of the coherent beamforming scheme and Alamouti based scheme in table 1 is upper bound, which is the capacity in noiseless at second hop, in order to express the capacity as closed form. To analyze DF relay case, a max-flow min-cut interpretation [5]. Roughly speaking, the rate of the information flow transmitted on the relay channel is constrained by the bottleneck corresponding to either the first cut (BS-relay link) or the second one (relay-mobile). For the coherent beamforming scheme and the Alamouti-based scheme, SNR at mobile can be evaluated easily without using upper bound unlike AF relay case. The process of detailed analysis is omitted and the approximation of outage probabilities for four schemes in high SNR is tabulated in table 1.

3.2 MIMO Relaying Schemes Using One Relay with Two Antennas

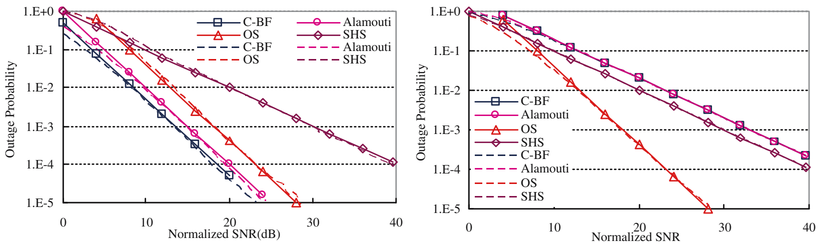
For AF relay transmission, two relays with one antenna scheme has the same performance of the capacity and outage probability as one relay with two antennas scheme when the path loss of geometry is not considered, since each antenna of a relay retransmits only an amplified version of each received signal. For DF relay transmission, the capacity of first hop channel is increased by array gain at receiver. Therefore unlike AF relay case, we can expect that the performance of one relay with two antennas is better than that of two relays with single antennas in terms of capacity outage. Table 1 denotes outage probabilities for 4 schemes. When long-term fading which includes path loss, shadowing is considered, we will show the result by simulation in chapter 4.

Table 1. Outage Performance

Scheme	2 AF relays with 1 antenna	2 DF relays with 1 antenna	1 DF relay with 2 antennas
Coherent beamforming	$\frac{1}{2\sigma^2_{br_1}\sigma^2_{br_2}}(\frac{2^{2R}-1}{SNR})^2$	$(\frac{1}{\sigma^2_{br_1}}+\frac{1}{\sigma^2_{br_2}})(\frac{2^{2R}-1}{SNR})$	$\frac{1}{2}(\frac{1}{\sigma^2_{br_1}\sigma^2_{br_2}}+\frac{1}{\sigma^2_{r_1m}\sigma^2_{r_2m}})$ $\cdot(\frac{2^{2R}-1}{SNR})^2$
Optimal selection	$(\frac{1}{\sigma^2_{br_1}}+\frac{1}{\sigma^2_{r_1m}})$ $\cdot(\frac{1}{\sigma^2_{br_2}}+\frac{1}{\sigma^2_{r_2m}})$ $\cdot(\frac{2^{2R}-1}{SNR})^2$	$(\frac{1}{\sigma^2_{br_1}}+\frac{1}{\sigma^2_{r_1m}})$ $\cdot(\frac{1}{\sigma^2_{br_2}}+\frac{1}{\sigma^2_{r_2m}})$ $\cdot(\frac{2^{2R}-1}{SNR})^2$	$(\frac{1}{2\sigma^2_{br_1}\sigma^2_{br_2}}+\frac{1}{\sigma^2_{r_1m}\sigma^2_{r_2m}})$ $\cdot(\frac{2^{2R}-1}{SNR})^2$
Alamouti-based	$\frac{1}{\sigma^2_{br_1}\sigma^2_{br_2}}(\frac{2^{2R}-1}{SNR})^2$	$(\frac{1}{\sigma^2_{br_1}}+\frac{1}{\sigma^2_{br_2}})(\frac{2^{2R}-1}{SNR})$	$\frac{1}{2}(\frac{1}{\sigma^2_{br_1}\sigma^2_{br_2}}+\frac{4}{\sigma^2_{r_1m}\sigma^2_{r_2m}})$ $\cdot(\frac{2^{2R}-1}{SNR})^2$
Second hop selection	$(\frac{\sigma^{-2}_{br_1}\sigma^{-2}_{r_2m}+\sigma^{-2}_{br_2}\sigma^{-2}_{r_1m}}{\sigma^{-2}_{r_1m}+\sigma^{-2}_{r_2m}})$ $\cdot(\frac{2^{2R}-1}{SNR})$	$(\frac{\sigma^{-2}_{br_1}\sigma^{-2}_{r_2m}+\sigma^{-2}_{br_2}\sigma^{-2}_{r_1m}}{\sigma^{-2}_{r_1m}+\sigma^{-2}_{r_2m}})$ $\cdot(\frac{2^{2R}-1}{SNR})$	$(\frac{1}{2\sigma^2_{br_1}\sigma^2_{br_2}}+\frac{1}{\sigma^2_{r_1m}\sigma^2_{r_2m}})$ $\cdot(\frac{2^{2R}-1}{SNR})^2$

4 Simulation Results

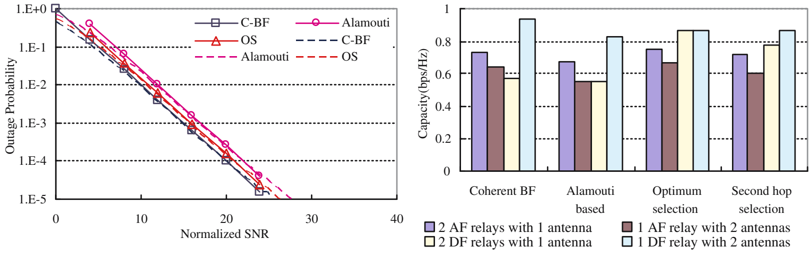
Fig. 2 shows the outage probability performance of AF and DF relay case in symmetric networks in which the fading variances are identical, e.g., $\sigma^{-2}_{br_i} = \sigma^{-2}_{r_im} = 1$. For AF relay, all schemes except the second hop selection scheme provide the second order diversity gain in high SNR. Even though the Alamouti-based scheme does not require feedback information, the second diversity is achieved. For DF relay, despite using more channel information, the coherent BF scheme does not offer the second order diversity. Since the relay received signal should be fully decoded at the relay, the relay channel capacity is limited by the capacity of first hop (BS-relay link). The coherent beamforming scheme has rather normalized SNR loss of 3dB than the second hop selection schemes which require only relay-mobile channel information in high SNR region.



(a) AF relay case

(b) DF relay case

Fig. 2. Outage probabilities of cooperative transmission



(a)

(b)

Fig. 3. (a) Outage probabilities of MIMO relay transmission (b) Average capacity

For AF relay transmission, the performance is identical between two relays with a single antenna and one relay with two antennas if we do not consider the path loss by effects due to the geometry. Fig. 3 (a) shows the outage probability performance of DF relay with two antennas in symmetric networks. In contrast to two relay with one antenna case, the coherent BF scheme and the Alamouti based scheme can offer the second order diversity since the capacity of BS-relay channel is increased by the array gain of relay antennas.

Fig. 3 (b) shows the average capacity which is obtained by system level simulation. We follow the evaluation methodology [9] submitted to 3GPP2 (3rd Generation Partnership Project 2) specification for the evaluation of cdma2000. The cell is divided by three sectors and only one sector is considered. Rayleigh fading channel based on Jake's model is considered and all channel elements are assumed to be independent. Mobile speed is fixed to 3 km/h.

5 Concluding Remarks

For AF relay, all schemes except the second hop selection scheme provide the second order diversity gain in high SNR region. Even though the Alamouti-based scheme does not require feedback information, the second diversity is achieved. For DF relay, despite using more channel information, the coherent BF scheme does not offer the second order diversity. Since the relay received signal should be fully decoded at the relay, the relay channel capacity is limited by the capacity of first hop (BS-relay link). The coherent beamforming scheme has rather normalized SNR loss of 3dB than the second hop selection schemes which require only relay-mobile channel information in high SNR region. A system level simulation is evaluated to analyze the effects of user distribution in terms of the average capacity. For AF relay, the capacity of the cooperative relaying model is higher than that of the MIMO relaying model as about 0.2 bps/Hz since AF relay can reduce the effect of path loss. For DF relay, the MIMO relaying model provides remarkable capacity gain about larger than 0.3 bps/Hz for the coherent beamforming and the Alamouti-based schemes since the effect of array gain at relay for these schemes is more dominant factor to increase the capacity compared with the effect of reducing path loss. For the selection combining-based schemes such the optimum selection scheme and the second selection scheme, the capacity of the cooperative relaying model and the MIMO relaying model are similar, since the effect of array gain is similar to that of reducing path loss.

References

1. H. Yanikomeroglu ”Fixed and mobile relaying technologies for cellular networks,” Second Workshop on Applications and Services in Wireless Networks (ASWN’02), pp. 75-81, 3-5 July 2002, Paris, France.
2. R. Bruno and M. Conti, ”Mesh networks: commodity multihop ad hoc networks”, IEEE Communications Magazine, vol. 43, vol. 3, pp. 123-131, March 2005.
3. J. Nicholas Laneman, David N. C. Tse, and Gregory W. Wornell ”Energy-efficient antenna sharing and relaying for wireless networks,” in Proc.of IEEE Wireless Commun. and Networking Conf., Chicago, IL, March 2000, vol. 1, pp. 7 - 12.
4. J. Nicholas Laneman, David N. C. Tse, and Gregory W. Wornell ”Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior,” IEEE Trans. Inform. Theory, vol. 50, no. 12, pp. 3062-3080, Dec. 2004.
5. T. M. Cover and J. A. Thomas, Elements of Information Theory. New York:Wiley, 1991.
6. P.A. Anghel, M. Kaveh, ”On the performance of distributed space-time coding systems with one and two non-regenerative relays,” IEEE Transactions on Wireless Communications, to appear in 2005.
7. P. A. Anghel, G. Leus, M. Kaveh, ”Multi-user space-time coding in cooperative networks,” in Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP’03), Hong Kong, April 6-10, 2003, vol. 4, pp. 73-6.
8. T. Lo. Maximal ratio transmission. IEEE Trans. Comm., 47(10), 1458-1461, October 1999.
9. 3GPP2 C.R1002-0 Version 1.0, ”1xEV-DV Evaluation Methodology, Revision 0,” December 10, 2004.

IP Mobility Support with a Multihomed Mobile Router

Hee-Dong Park[1], Dong-Won Kum[2], Yong-Ha Kwon[2],
Kang-Won Lee[2], and You-Ze Cho[2]

[1]Department of Computer Engineering, Pohang College, Pohang, 791-711, Korea
hdpark@pohang.ac.kr
[2]School of Electrical Engineering & Computer Science,
Kyungpook National University, Daegu, 702-701, Korea

Abstract. This paper proposes a multihoming-based seamless handover scheme using a mobile router with dual egress interfaces for wireless train networks. The proposed scheme deploys dual antennas which are individually located at each end of the train for space diversity and connected to each egress interface of a mobile router. Since one of the two egress interfaces of the mobile router can continuously receive packets through its antenna while the other is undergoing a handover, the proposed scheme can support a seamless handover providing no service disruption or packet loss.

1 Introduction

Network mobility (NEMO) basic support is concerned with managing the mobility of an entire network [1]. Public transportation, such as trains and buses, is an example of the mobile networks [2]. The NEMO basic protocol will be built on Mobile IPv6 with minimal extensions [3]. Therefore, the handover mechanism of a mobile router (MR) is essentially the same as that of a mobile node (MN) with Mobile IP. Recently, various multihoming issues have been presented in the NEMO Working Group. The multihoming is necessary to provide constant access to the Internet and to enhance the overall connectivity of hosts and mobile networks [4][5]. This requires the use of several interfaces and technologies since the mobile network may be moving in distant geographical locations where different access technologies are provided. The additional benefits of the multihoming are fault tolerance/redundancy, load sharing, and policy routing. However, there is no requirement or protocol defining how to use several interfaces with a mobile network. This paper proposes a multihoming-based handover scheme using an MR with dual egress interfaces, which cooperate with each other to perform seamless handovers for a large moving network, such as trains. The proposed scheme deploys dual antennas which are individually located at each end of the moving network for space diversity and connected to each egress interface of the MR. One of the two egress interfaces can continuously receive packets through its antenna, while the other is undergoing a handover. This can support a seamless handover providing no service disruption or packet loss.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1144–1149, 2006.

2 Multihoming-Based Seamless Handover

The proposed system is assumed to be deployed in a large mobile network such as a train. Fig. 1 shows the vehicle network structure of the proposed scheme. For multihoming, an MR with dual antennas (Head_ANT and Tail_ANT) can be deployed in the vehicle. The Head_ANT and Tail_ANT are located in the front and back end of the train, respectively. The multihomed MR has at least two egress interfaces connected to the dual antennas, and each of the two interfaces has its own HoA and CoA. In the proposed scheme, the terms Head_CoA and Head_HoA are used to represent the CoA and HoA of the interface connected to the Head_ANT, while the terms Tail_CoA and Tail_HoA are used for the interface connected to the Tail_ANT. Also, Mobile IPv6 is assumed to be used for the proposed system. There are APs in each car of the train, and they are connected to the MR through a switch.

Fig. 2 shows the handover procedure of the proposed scheme. When both antennas stay in the Old_AR's coverage area, the MR communicates with the Old_AR through the Tail_ANT, while the Head_ANT waits for an impending handover.

① Phase 1: As the mobile network moves, the Head_ANT reaches New_AR's coverage area prior to the Tail_ANT. After the MR receives the network prefix information from the New_AR through the Head_ANT and associates with the New_AR by creating a CoA (Head_CoA), it sends a proxy BU message to the HA. The proxy BU message contains the new Head_CoA and the Tail_HoA instead of the Head_HoA. This makes the HA to be under the illusion that the MR has only one egress interface, and prevents the HA from having multiple bindings. The Tail_ANT, however, actually continues to receive packets in the Old_ARs coverage area, thus packet loss can be prevented. After receiving the Proxy BU message, HA updates the binding and delivers packets to the MR through the New_AR. When the MR receives a Proxy BU ACK message from the HA through the Head_ANT, it sends the data packets originated from the MNNs to the Internet through the Head_ANT.

② Phase 2: When the Tail_ANT stays in the Old_AR's coverage area and the Head_ANT stays in the New_AR's coverage area simultaneously, the MR can send and receive data packets through the Head_ANT, and it may also receive in-transit data packets destined to the Old_AR through the Tail_ANT.

③ Phase 3: If the MR receives router advertisement messages from the New_AR through the Tail_ANT, the MR performs a handover. At this time, the MR

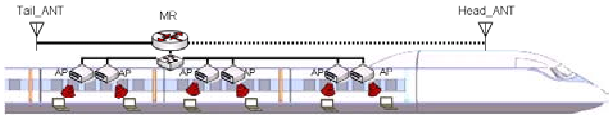


Fig. 1. Vehicle network structure of the proposed scheme

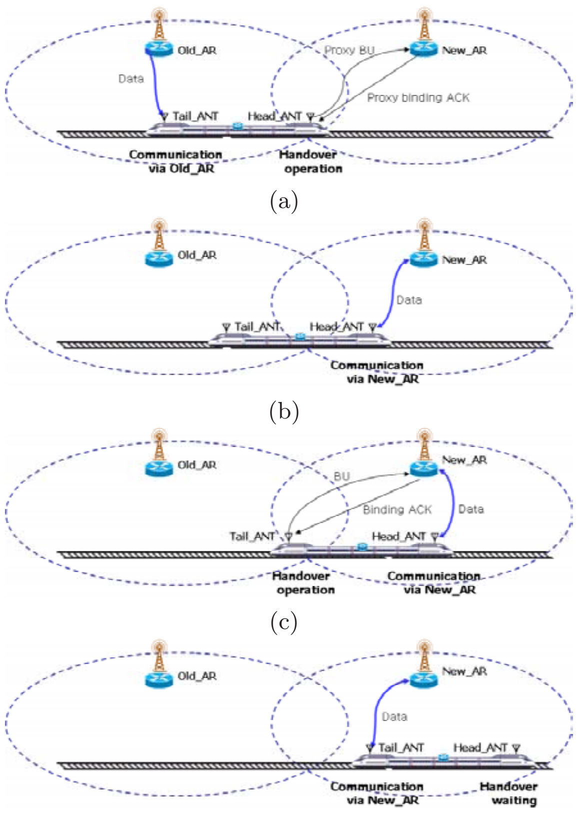


Fig. 2. Handover procedure of the proposed scheme (a)Phase 1, (b)Phase 2, (c)Phase 3, and (d)Phase 4

sends a general BU message including the Tail_CoA and Tail_HoA through the New_AR. After receiving a Binding ACK message, the MR can send and receive packets through the Tail_ANT.

④ Phase 4: When both antennas stay in the New_AR's coverage area, the MR communicated with the New_AR through the Tail_ANT, while the Head_ANT waits for an impending handover again.

In the proposed scheme, the proxy BU and the proxy BU ACK messages are newly introduced. The formats of these messages, however, are the same as those of the general BU and BU ACK messages in Mobile IPv6. The only difference between the proxy BU message and the general BU message is about the content of the messages. That is, the MR inserts the Tail_HoA into the Proxy BU message in place of Head_HoA. Table 1 shows the binding information maintained in the HA.

Table 1. Binding information in the HA

Binding Phases	HoA	CoA
Phase 1	Tail_HoA	new Head_CoA
Phase 2	Tail_HoA	new Head_CoA
Phase 3	Tail_HoA	new Tail_CoA
Phase 4	Tail_HoA	new Tail_CoA

3 Performance Evaluation

Two critical performance issues are service disruption time and packet loss during handovers. For performance analysis, we use parameters as follows: total handover latency (T_{HO}), movement detection delay (T_{MD}), CoA configuration delay ($T_{CoA-Conf}$), delay for BU (T_{BU}), router advertisement interval (τ), round-trip time between MR and AR (RTT_{MR-AR}), and round-trip time between AR and HA (RTT_{AR-HA}). In this paper, we regard the service disruption time as the total handover latency, T_{HO}. The total handover latency the NEMO basic solution is given by:

$$\begin{aligned} T_{HO} &= T_{MD} + T_{CoA-conf} + T_{BU} \\ &= 2\,\tau + 2RTT_{MR-AR} + RTT_{AR-HA} \end{aligned} \tag{1}$$

Since packet loss does not occur during the time when the CN traffic travels from the HA to an MR after the completion of the BU, the packet loss period (T_{loss}) during a handover can be expressed as T_{HO} - 0.5 RTT_{MR-HA}.

Packet loss ratio (ρ_{loss}) is defined as the ratio of the number of lost packets during a handover to the total numbers of transmission packets in a cell. This can be also expressed as:

$$\rho_{loss} = \frac{T_{loss}}{T_{cell}} \times 100 \quad (\%) \tag{2}$$

where T_{cell} is the time it takes an MR to pass through a cell.

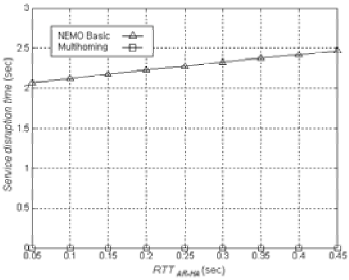


Fig. 3. Service disruption time

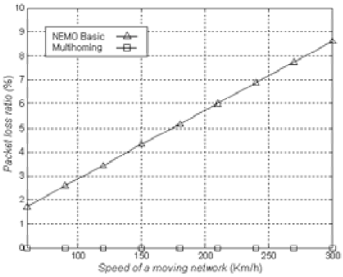


Fig. 4. Packet loss ratio

However, in the proposed scheme, handovers of the Head_ANT and the Tail_ANT alternate with each other, thereby the total service disruption time and packet loss will be zero. Fig. 3 and 4 compare the service disruption time and packet loss ratio between the proposed scheme and the NEMO basic support, respectively. We assume that the router advertisement interval is 1 second, the radius of AR cell coverage is 1 km, and RTT_{MR-AR} is 10 msec. RTT_{AR-HA} is assumed to be 100 msec in Fig. 4. As shown, the service disruption time and packet loss ratio of the proposed scheme will be zero.

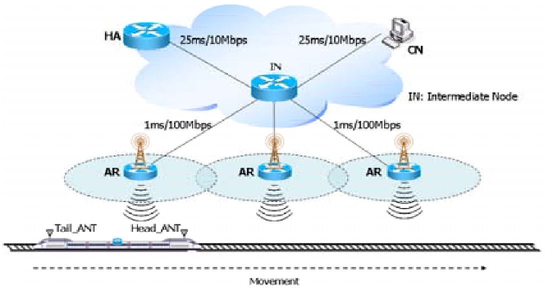
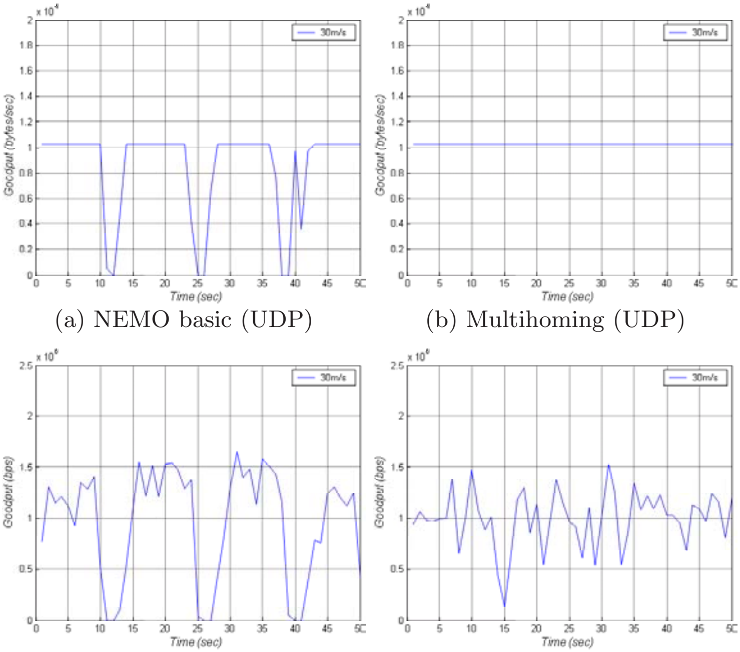


Fig. 5. Network model for simulation



(a) NEMO basic (UDP)

(b) Multihoming (UDP)

(c) NEMO basic (TCP)

(d) Multihoming (TCP)

Fig. 6. Comparison of the UDP and TCP goodput behaviors at the speed of 30 m/sec

Fig. 5 shows the network model for simulation: Coverage radius of an AR is 250m, distance between ARs is 400m, router advertisement interval is 1sec, IEEE 802.11b as the wireless LAN, and distance between dual antennas is 200m. We have simulated for two traffic types: UDP and TCP. For UDP, the 512-byte packets were sent repeatedly at a constant rate of 20 packets per second from the CN to a mobile network node (MNN) residing in the train. For TCP, FTP traffic was generated with a full window. Fig. 6 compares the UDP and TCP goodput behaviors between the proposed scheme and the NEMO basic, respectively. From this figure, we note that the proposed scheme can provide a higher goodput in both cases of the UDP and the TCP, because the proposed scheme has no service disruption during handovers.

4 Conclusion

This paper proposed a seamless handover scheme using a multihomed MR with dual antennas for trains. Each of the dual antennas is located at each end of a mobile network for space diversity. One of the two egress interfaces of the MR can continuously receive packets through its antenna, while the other is undergoing a handover. Therefore, the proposed scheme can provide no service disruption or packet loss during handovers. However, the proposed scheme has some overhead in comparison with NEMO basic support. The overhead involves the cost to maintain dual MRs with additional signaling messages.

Acknowledgment

This work was supported in part by the KOSEF (contract no.: R01-2003-000-10155-0) and the ITRC of the Ministry of Information and Communication (MIC), Korea.

References

1. V. Devarapalli et al., “Nemo basic support protocol,” *IETF RFC* 3963, Jan. 2005.
2. E. K. Paik and Y. H. Choi, “Prediction-Based Fast Handoff for Mobile WLANs,” in *Proc. of ICT*, vol. 1, pp. 748-753, Feb. 2003.
3. D. Johnson et al., “Mobility Support in IPv6,” *IETF RFC* 3775, June. 2004.
4. C. Ng, J. Charbon, E. K. Paik, and T. Ernst, “Analysis of multihoming in network mobility support,” *InternetDraft*, Feb. 2005.
5. N. Montavont, T. Ernst, and T. Noel, “Multihoming in nested mobile networking,” *SAINT2004Workshops*, Jan. 2004.

Performance Analysis and Design: Power Saving Backoff Algorithm for IEEE 802.11 DCF*

Feng Zheng, Barry Gleeson, and John Nelson

Department of Electronic and Computer Engineering,
University of Limerick, Limerick, Ireland
feng.zheng@ul.ie, barry.gleeson@ul.ie, john.nelson@ul.ie

Abstract. In this paper, we propose an approach to saving power by more aggressively using sleep mode. The sleep duration is estimated by measuring the on-line traffic information, i.e. slot utilization. A considerable amount of power can be saved by using the approach. Our approach does not use any additional signaling channel, nor does it need any overhead in the involved protocol. The algorithm is readily implementable, requiring minimum processing and memory resources.

Keywords: Wireless networks, 802.11, power saving, sleep mode, performance evaluation, distributed coordination function (DCF).

1 Introduction

Wireless hosts are usually powered by batteries which provide a limited amount of energy. Therefore, techniques to reduce energy consumption are of interest. One way to conserve energy is to suitably adjust transmit power to reduce energy consumption. Another alternative is to use power saving mechanisms, which allow a node to enter a doze state by powering off its wireless network interface when deemed reasonable [5, 6, 7]. The objective of this paper is to study the power saving technique for IEEE 802.11 wireless networks in the context of the latter approach.

The IEEE 802.11 power saving mechanism is based on the idea of reservation, where all nodes book their transmit requests and schedules and listen to the receive request during specific reservation intervals. All nodes not participating in transmission or reception of packets go into doze mode until the next reservation period. But to book the requests, every node should wake up periodically, which still consumes energy and is often not necessary. Considering the fact the idle state occupies quite a lot of time, it is desirable to switch the contending stations into the doze state in the backoff stage. In [4], a new scheme is proposed to implement this idea, where the sleep time is calculated based on the statistics of the channel. In this paper, we will provide a detailed theoretical analysis for the sleep algorithm and propose a new estimate for the sleep duration, which is

* This work was supported by National Communications Network Research Centre, a Science Foundation Ireland Research Project.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1150–1156, 2006.

based on the real time traffic information of the channel. Our mechanism does not use any additional signaling channel, nor does it need any overhead in the involved protocol.

2 IEEE 802.11 MAC and Sleep Algorithm

IEEE 802.11 employs a CSMA/CA MAC protocol with binary exponential backoff, referred to as the distributed coordination function (DCF), to access the medium, the details of which have been summarized in [1].

In a typical usage scenario, a contending station spends most of its time to listen the channel. Therefore the station can be pushed into sleep in the process of decreasing its backoff time counter (BTC). Let us denote the backoff time counter as $b(t)$ for a given station, where the time t has been discretized by slot time. For the moment, we assume that the sleep duration is a constant L when the backoff time counter $b(t)$ is larger than or equal to L. Now the sleep algorithm can be described as follows:

If $b(t) \geq L$, then the station will be switched into the sleep state for L slot time. When it wakes up, its backoff time counter will be reduced by L. If $b(t) < L$, then the station will be switched into a sleep state for $b(t)$ slot time. When it wakes up, its backoff time counter will be set to zero. Therefore the sleep duration of a station will be $\min\{L, b(t)\}$.

From the above sleep algorithm, we can see that the station in the sleep mode will not wake up to sense the channel even if it is time to do it, i.e., the station will not wake up periodically during its sleep time. This is the basic difference between our sleep mechanism and the standard power saving mechanism of IEEE 802.11.

3 Throughput Analysis and Estimation of Sleep Length

3.1 Stationary Probability for the Case $L \leq W_0$

An analytical model for the standard backoff mechanism of the IEEE 802.11 MAC has been developed by Bianchi [1]. The analysis of our sleep algorithm is mainly based on Bianchi's model. By introducing the process of backoff stage $s(t)$, the bi-dimensional process $\{s(t), b(t)\}$ will become a Markov chain [1]. Its state transition probabilities, considering sleep, are illustrated in Fig. 1.

In this model, we assume that, at each transmission attempt, and regardless of the number of retransmissions suffered, each packet transmission collides with constant and independent probability. In Fig. 1, the states marked with $(i, 0)$ stands for those states at which the station is ready to transmit data packets; the states marked with γ are intermediate states that will transfer into sleep states; the states marked with α are states at which the station will sleep for a slot time; the states marked with β are awake states at which the station will sense whether or not the channel is idle and decrease its backoff counter if yes. Note that when a station is transferred to a state $\alpha(i, k)$, it will sleep for k time slots before transferring to the state $(i, 0)$ to transmit data packets. Denote

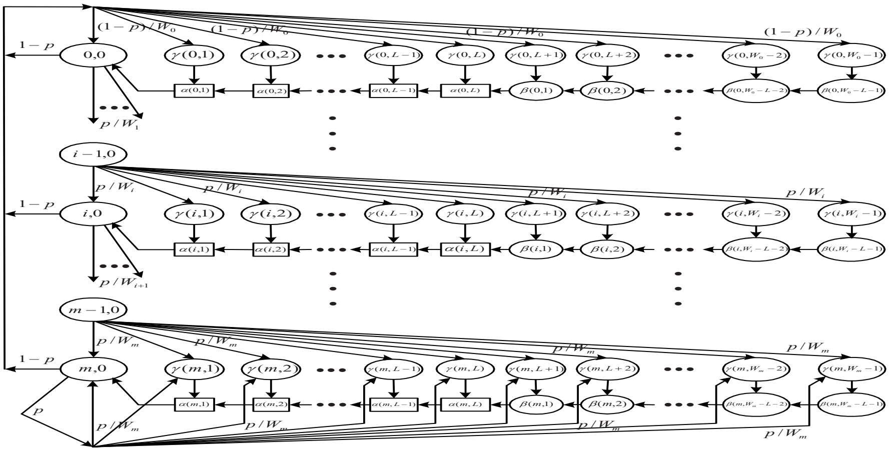


Fig. 1. Markov chain model for the backoff counter with sleep mechanism for the case of $L \leq W_0$. In this figure, the probabilities of those state transitions which are not marked are all one.

by $p_{\mathtt{state}}$ the stationary probability of a state, and by τ the probability that a station transmits in a randomly chosen slot time. Then it can be found that

$$p_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W_0+1)+pW_0(1-(2p)^m)}, \ \tau = \frac{2(1-2p)}{(1-2p)(W_0+1)+pW_0(1-(2p)^m)}.$$

3.2 Stationary Probability for the Case $L > W_0$

In this case, we suppose that there is an integer ν such that $W_\nu - 2 < L \leq W_{\nu+1} - 2$. Due to space limit, the state transition diagram for this case is not illustrated. The following can be obtained:

$$\begin{aligned} p_{0,0} &= \left\{ \frac{1}{1-p}\left[L + \frac{5}{2} - (L+2)p^{\nu+1} + \frac{W_0}{2}(2p)^m\right] - \frac{L+1}{W_0}\cdot\frac{1-(p/2)^{\nu+1}}{1-p/2}\right. , \\ &\left. + \frac{W_0}{1-2p}\left[(2p)^{\nu+1} - \frac{1}{2} - \frac{1}{2}(2p)^m\right]\right\}^{-1} \\ \tau &= \left\{ L + \frac{5}{2} - (L+2)p^{\nu+1} + \frac{W_0}{2}(2p)^m - \frac{L+1}{W_0}\cdot\frac{(1-(p/2)^{\nu+1})(1-p)}{1-p/2}\right. \\ &\left. + \frac{1-p}{1-2p}\left[(2p)^{\nu+1} - \frac{1}{2} - \frac{1}{2}(2p)^m\right]W_0\right\}^{-1} . \end{aligned} \quad (1)$$

To find the transmission probability, we need to calculate p. Suppose there are n contending stations. Then following the same argument as in [1], we have

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

Combining equation (2) with (3.1) or (1), we can solve p and then τ.

3.3 Calculation of Throughput and Estimation of Sleep Duration

To determine the sleep duration and calculate the throughput, it is convenient to use the concept of virtual transmission time, as introduced by Calì et al [3]. A virtual transmission time is the time interval between two successful transmissions, which includes a successful transmission and may include several collision intervals (see Fig. 2 of [3]). Define N_i to be the number of consecutive idle slots and N_c to be the number of collisions in a virtual transmission time. Assume that at any a given time slot, whether a station transmits is independent of any other stations and that the events that whether or not a station will transmit at two different time slots are independent. Consider one transmission attempt in one virtual transmission time. Then we can find that

$$\mathtt{Pr}\{N_i = k\} = [1-(1-\tau)^n]\cdot[(1-\tau)^n]^k\,, \quad \mathcal{E}(N_i) = \frac{(1-\tau)^n}{1-(1-\tau)^n}, \qquad (3)$$

where $\mathtt{Pr}$ denotes the probability of an event, and $\mathcal{E}$ denotes the expectation of a random variable. Let P_C denote the probability that the transmissions of the n contending stations are colliding, and P_S denote the probability that a transmission of one of the n contending stations is successful. Then we have

$$P_S = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}, \quad P_C = \frac{1-(1-\tau)^n - n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}.$$

Therefore

$$\mathtt{Pr}\{N_c = k\} = P_C^k P_S, \quad \mathcal{E}(N_c) = \sum_{k=1}^{\infty} k P_C^k P_S = \frac{1-(1-\tau)^n - n\tau(1-\tau)^{n-1}}{n\tau(1-\tau)^{n-1}}.$$

Let l_{pac} be the average packet payload size (in time slot), T_s the average time the channel is sensed busy because of a successful transmission, and T_c the average time the channel is sensed busy by each station during a collision, and σ the duration of an empty slot time. The average time which the channel spends on colliding transmissions and idle before a successful transmission is thus given by

$$T_{c\&i} = \mathcal{E}(N_c)T_c + (\mathcal{E}(N_c)+1)\mathcal{E}(N_i)\sigma = \left[\frac{1}{n\tau(1-\tau)^{n-1}} - \frac{1-\tau}{n\tau} - 1\right]T_c + \frac{1-\tau}{n\tau}\sigma. \quad (4)$$

The throughput of the channel reads as $\rho = \frac{l_{\text{pac}}}{T_s + T_{c\&i}}$.

Equation (4) motivates us to adopt the sleep duration as

$$T_{\text{sleep}} = \mu T_{c\&i} = \mu\left\{\left[\frac{1-(1-\tau)^n}{n\tau(1-\tau)^{n-1}} - 1\right]T_c + \frac{1-\tau}{n\tau}\sigma\right\}. \qquad (5)$$

where the parameter μ is introduced to take into account the effect of the number of contending stations on the duration of the sleep. The duration T_{sleep} chosen according to (5) is based on the idea that the station in a backoff state will stay in sleep and wake up at the moment that its transmit attempt will be probably most successful. It is reasonable to choose $\mu = \frac{1+n}{2}$. Since the BTC will be frozen when a channel is sensed busy, the sleep duration as counted in backoff time counter will be accordingly

$$L = \left\lfloor \mu \left\{ \left[\frac{1-(1-\tau)^n}{n\tau(1-\tau)^{n-1}} - 1 \right] + \frac{1-\tau}{n\tau} \right\} \right\rfloor = \left\lfloor \mu \left\{ \frac{1}{n\tau(1-\tau)^{n-1}} - 1 \right\} \right\rfloor . \quad (6)$$

The algorithm (5) or (6) is difficult to implement since it requires a station to have the knowledge about the number of the potential contending stations. However, in some cases, we can simplify the matter. This is the case where $\tau \ll 1$.

Similar to [2], define the slot utilization as follows

$$U_s = \frac{\texttt{Number\_Busy\_Slots}}{\texttt{Number\_Available\_Slots}}, \quad (7)$$

where `Number_Busy_Slots` is the number of slots in the backoff interval in which one or more stations start a transmission attempt (a transmission attempt can be either a successful transmission or a collision); and `Number_Available_Slots` is the total number of slots available for transmission in the backoff interval, i.e., the sum of idle and busy slots. It can be shown, in the case of $\tau \ll 1$, that

$$T_{\text{sleep}} \approx \mu \left[\frac{(n-1)\tau}{2(1-\tau)} T_c + \frac{1-\tau}{n\tau}\sigma \right] \approx \mu \left(\frac{U_s-\tau}{2} T_c + \frac{1}{U_s}\sigma \right), \quad (8)$$

$$L \approx \left\lfloor \mu \left(\frac{1}{U_s} - 1 \right) \right\rfloor, \quad \mu \approx \frac{1}{2} + \frac{U_s}{2\tau}. \quad (9)$$

Now the sleep algorithm can be summarized as follows:

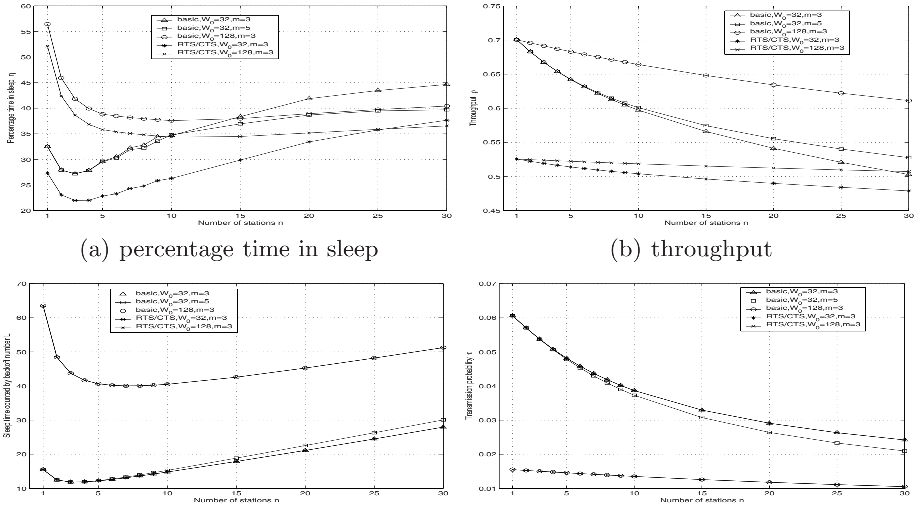
Algorithm 1

- Step 1: Measure U_s and τ. Calculate T_{sleep} and L according to (8) and (9).
- Step 2: If $b(t) \geq L$, then the station is switched into sleep for a period of T_{sleep}; when waking up, its BTC is reduced by L, i.e., $b(t) - L \rightarrow b(t)$.
  If $b(t) < L$, then the station is switched into sleep for a period of $T_{\text{sleep}} \cdot \frac{b(t)}{L}$; when waking up, its BTC is set to zero, i.e., $0 \rightarrow b(t)$.
- Step 3: Sense the channel and transmit correspondingly. Goto step 1.

4 Numerical Results

The values of the parameters used in the numerical results are as follows: packet payload = 8184 bits, MAC header = 272 bits, PHY header = 128 bits, ACK = 112 bits + PHY header, RTS = 160 bits + PHY header, CTS = 112 bits + PHY header, channel bit rate = 11 Mbit/s, propagation delay = 1 μs, slot time = 50 μs, SIFS = 28 μs, DIFS = 128 μs.

Fig. 2 (b) depicts the throughput of the network executing the sleep algorithm. As is shown, the throughput of the network is similar to the one using the IEEE 802.11 standard MAC. This is because the sleep time L, counted by backoff number, is less than W_0, as illustrated in Fig. 2 (c) , for all the cases studied



(a) percentage time in sleep

(b) throughput

(c) sleep time counted by backoff number

(d) transmission probability

Fig. 2. The performance of the 802.11 network executing the sleep algorithm

here. Thus the transmission probability of a station at any time slot is the same as the one of the network employing the standard MAC. The percentage time that a station spends in sleep is illustrated in Fig. 2 (a). From this figure we can see that, by using the sleep algorithm developed here, a considerable amount of energy can be saved for networks adopting IEEE 802.11 MAC. The transmission probability of a station is plotted in Fig. 2 (d). It is seen that the probability τ is much less than one for all the cases investigated here. Therefore, it is justified that the sleep duration can be estimated, based on equation (9), by measuring the on-line traffic information U_s.

Finally, we point out that it consumes power for transceivers to enter and exit sleep, which has not been considered in this paper. This remains the future topic to improve the algorithm developed here. Also note that when n further increases, L may be larger than W_0. In this case, the throughput of the channel might decrease.

References

1. G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Select. Areas Commun.*, 18:535-547, 2000.
2. L. Bononi, M. Conti, and E. Gregori. Runtime optimization of IEEE 802.11 wireless LANs performance. *IEEE Trans. Parallel and Distributed Systems*, 15:66-80, 2004.
3. F. Calì, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE Trans. Networking*, 8:785-799, 2000.
4. B. Gleeson and J. Nelson. PSBP: Power saving backoff prediction in IEEE 802.11. submitted for publication, 2005.

5. E.-S. Jung and N. H. Vaidya. An energy efficient MAC protocol for wireless LANs. In *Proc. INFOCOM 2002*, pp.1756-1764, 2002.
6. S. Singh and C. S. Raghavendra. PAMAS power aware multi-access protocol with signalling for ad hoc networks. *Computer Comm. Review*, pp.5-26, July 1998.
7. H. Woesner, J.-P. Ebert, M. Schläger, and A. Wolisz. Power-saving mechanisms in emerging standards for wireless LANs: The MAC level perspective. *IEEE Personal Communications*, pp.40-48, June 1998.

A Fast Pattern-Matching Algorithm for Network Intrusion Detection System*

Jung-Sik Sung[1], Seok-Min Kang[2], and Taeck-Geun Kwon[2,**]

[1] ETRI, 161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea
jssung@etri.re.kr
[2] Chungnam National University, 220 Gung-dong, Yuseong-gu, Daejeon, 305-764, Korea
{esemkang, tgkwon}@cnu.ac.kr

Abstract. We present a multi-gigabit rate multiple pattern-matching algorithm with TCAM that enables protecting against malicious attacks in a high-speed network. The proposed algorithm significantly reduces the number of TCAM lookups per payload with *m*-byte *jumping window* scheme. Due to the reduced number of TCAM lookups, we can easily achieve multi-gigabit rate for scanning the packet payload in order to inspect the content. Furthermore, multi-packet inspection is achieved easily by the extended state transition diagram with the *shifting distance*. With experimental results, we have clearly justified the proposed algorithm works well for a multi-gigabit network intrusion detection system.

1 Introduction

Network intrusion detection systems (NIDSs) monitor every packet in the network to detect malicious attacks. In a high-speed network, an NIDS may be overloaded as the packet arrival rate becomes high. Hence, the hardware-based approach of implementing the NIDS will be appropriate in order to support the high-speed network. Some researches [1], [2], [3] focus on the hardware implementation to achieve the line-speed intrusion detection. Recently, technologies for high performance network processors have driven a new breed of solutions that perform at high data rates while remaining flexible through software [4].

There are many approaches for solving multiple pattern-matching problems. The multiple pattern-matching algorithms [5], [6] use software approaches. However, software-based pattern-matching is not able to inspect all packets in the high-speed network. Gigabit rate pattern-matching algorithms such as [7], [8] are TCAM-based algorithms that can be used with TCAM. In this paper, the scheme in [7], [8] is referred to a '*sliding window*' in which every one-byte shifted fixed-length partial payload should be examined to match the TCAM and the partial payload should be extracted with a sliding window manner. Although the sliding window based pattern matching is intuitive and simple, the scheme has three problems. First, it has lower

* This research was supported in part by ITRC program of the Ministry of Information and Communication, Korea.

** Corresponding author.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1157 – 1162, 2006.

scan speed. It can provide an answer for searching a packet of length n, in a deterministic time of $O(n)$ TCAM lookups, because one TCAM lookup is needed for every byte position in the packet. Since the TCAM lookup time is known and fixed, we need to minimize the number of TCAM lookups per packet to support the multi-gigabit rate NIDS. Second, it is very complicated and needs more memory when the pattern is longer than TCAM width. Suppose the width of the TCAM is w bytes and let $T = t_0,t_1,\ldots,t_{n-1}$ be the text. Partial pattern is matched at $t_i,\ldots,t_{i+w-1}$, it should check whether occurrence of previous partial matching at $t_{i-w},\ldots,t_{i-1}$, and keeps matching information for next partial matching. Third, it does not support multi-packet inspection where the pattern split into continuous two payloads. This situation is common in NIDS, where intrusion signatures can be segmented into packets which contain the user data such as E-mails, attached files, etc.

In this paper, we revise deep packet inspection algorithm introduced in our recent paper [9] and extend the algorithm to provide content inspection over multiple packets. In our algorithm, TCAM lookups for searching a packet of length n, is $O(n/m)$, if the size of the jumping window is m. We devise the state transition diagram for keeping previous partial matching when the pattern is longer than TCAM width. So it is very simple and does not waste memory. In order to support multi-packet inspection, we use extended state transition diagram by alignment of the last jumping window of previous payload. In addition, we have implemented the proposed algorithm using Intel IXP28XX network processors (NPs) with TCAM. We have some preliminary experimental results which verify the proposed scheme improves significantly the performance of deep packet inspection.

The rest of the paper is organized as follows. In Section 2, we describe problems of multiple pattern-matching using TCAM. We explain the jumping window algorithms to map the multiple patterns into TCAM and efficiently scan packets at high speeds in Section 3. In Section 4, we extend the algorithm in order to inspect multiple packets. In Section 5, we give experimental results with our 10Gbps network processor based NIDS. Finally, we conclude the paper.

2 Jumping Window Pattern Match Algorithm

A pattern is a string to be searched for a payload and it usually appears at an arbitrary position in the payload. For example, virus and worm patterns are located in an attached file and they may appear at any position in the packet payload. We should search several sub-patterns relevant to each jumping window for matching a pattern if the pattern would be occupied into continuous several jumping windows of a payload. In other words, we can create TCAM entries, all possible position-aware patterns (PAPs) from one pattern. Therefore, we succeed pattern-matching with jumping-window scheme although the pattern appears at an arbitrary position in the payload.
Let $T = t_0,t_1,\ldots,t_{n-1}$ be the payload, and its length be n bytes. Let $P = p_0,p_1,\ldots, p_{m-1}$ be the pattern to be searched, and its length be m bytes. P is located in the substring of T, where

$$t_s,\ldots,t_{s+m-1} = p_0,\ldots,p_{m-1},\ 0 \le s \le n\text{-}m. \tag{1}$$

The payload is divided into multiple jumping window substrings with m-byte window, where

$$t_{sm},\ldots,t_{(s+1)m-1}, 0 \le s \le \left\lceil \frac{n-m}{m} \right\rceil . \tag{2}$$

If (m-i) sequential bytes of P is found from i^{th} position within s^{th} jumping window substring of T, where

$$t_{sm+i},\ldots,t_{(s+1)m-1} = p_0,\ldots,p_{m-1-i},\ i=0,1,2,\ldots,m\text{-}1. \tag{3}$$

Then, the rest of P, the remaining i bytes is found from the first position within $(s+1)^{th}$ jumping window substring of T. On this occasion, P is found in T. Therefore, we can make PAPs from P with 0~(m-1) shifting and can split PAPs into fixed-size sub-patterns, position-aware sub-patterns(PASes) as shown in Fig. 1 ('–' denotes "don't care" state of TCAM). We can match the pattern P in the payload T with jumping window scheme when these PASes generated from P are stored in the TCAM.

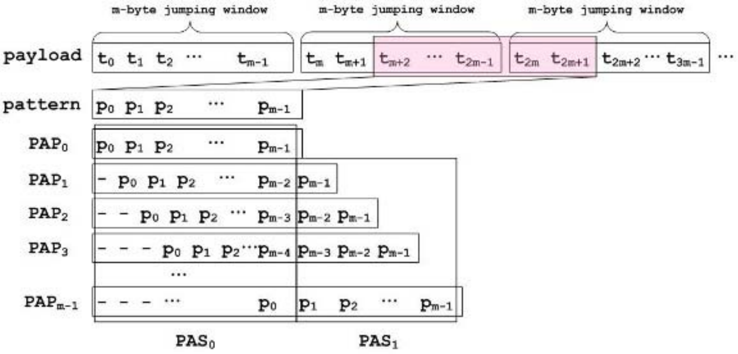

m-byte jumping window
m-byte jumping window
m-byte jumping window
payload
t_0 t_1 t_2 ··· t_{m-1}
t_m t_{m+1} t_{m+2} ··· t_{2m-1}
t_{2m} t_{2m+1} t_{2m+2} ··· t_{3m-1} ···
pattern
p_0 p_1 p_2 ··· p_{m-1}
PAP$_0$
PAP$_1$
PAP$_2$
PAP$_3$
PAP$_{m-1}$
- p_0 p_1 p_2 ··· p_{m-2} p_{m-1}
- - p_0 p_1 p_2 ··· p_{m-3} p_{m-2} p_{m-1}
- - - p_0 p_1 p_2···p_{m-4} p_{m-3} p_{m-2} p_{m-1}
- - - ··· p_0 p_1 p_2 ··· p_{m-1}
PAS$_0$
PAS$_1$

Fig. 1. Position-aware patterns and position-aware fixed sub-patterns

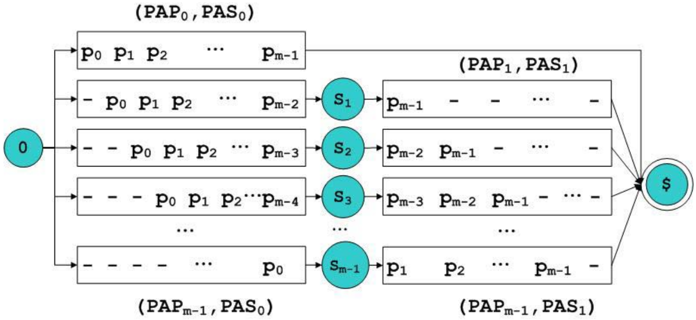

(PAP$_0$, PAS$_0$)
(PAP$_1$, PAS$_1$)
0
s_1
s_2
s_3
s_{m-1}
$
p_0 p_1 p_2 ··· p_{m-1}
- p_0 p_1 p_2 ··· p_{m-2}
p_{m-1} - - ··· -
- - p_0 p_1 p_2 ··· p_{m-3}
p_{m-2} p_{m-1} - ··· -
- - - p_0 p_1 p_2···p_{m-4}
p_{m-3} p_{m-2} p_{m-1} - ··· -
- - - - ··· p_0
p_1 p_2 ··· p_{m-1} -
(PAP$_{m-1}$, PAS$_0$)
(PAP$_{m-1}$, PAS$_1$)

Fig. 2. State transition diagram

Given the pattern of "GATT" the position of the pattern in the payload is one of "GATT," "–GATT," "––GATT," …, "(m -1)–GATT." When m is 4, the pattern may be found at the different position of the payload such as "GATT," "–GATT," "––GATT", or "–––GATT." We put the above derived patterns into the TCAM table.

Then, a TCAM lookup operation is carried out for every segment of *m* bytes called a jumping window for a packet payload. Usually, the width of the TCAM, which will be used for matching the pattern in a parallel way, is fixed. Therefore, if the TCAM width is smaller than the pattern, we have to split a long pattern into shorter sub-patterns with the same length of the TCAM width. If one PAP splits into several PASes, a pattern-matching operation will be completed when all PASes are matched to the TCAM entries in series. Hence, for the matching operation of multiple PASes, a PAS matching function requires the result of the previous PAS matching operation. To increase the speed of searching, we employ the state transition diagram to find the result of the previous PAS matching operation as shown in Fig. 2.

3 Multi-packet Inspection

We consider a pattern that split on two payloads T_i, T_{i+1} and it usually appears at several jumping windows of T_i, T_{i+1}. That is, it split into tail end of the former and beginning of the latter. In case of Fig. 3 (a), the pattern $P = p_0,p_1,\ldots,p_{m-1}$ split into "p_0" and "$p_1p_2\ldots p_{m-1}$" in T_i and T_{i+1}, respectively. At the window before last of T_i as illustrated in Fig. 3 (a), pattern-matching does not occur and the state is initial state, 0. We fit the last window with shifting distance because the remaining bytes are too small to fit into the jumping window. There is no previous pattern-matching, the state of the last window is initial state, 0. However in case of Fig. 3 (b), the partial pattern "p_0" is matched and transits to state S_{m-1}. Due to lookup TCAM with *m*-byte window, the last window of T_i needs *m*-byte alignment. The start point of last window is shifted to *m*-(*n*%*m*) bytes left. We call it *shifting distance*. In order to fit the last window, the pre-condition state should be changed according to the shifting distance. For example, state S_{m-1} must move into initial state 0 if 3 bytes, i.e., "—p_0," are shifted for fitting the last search window.

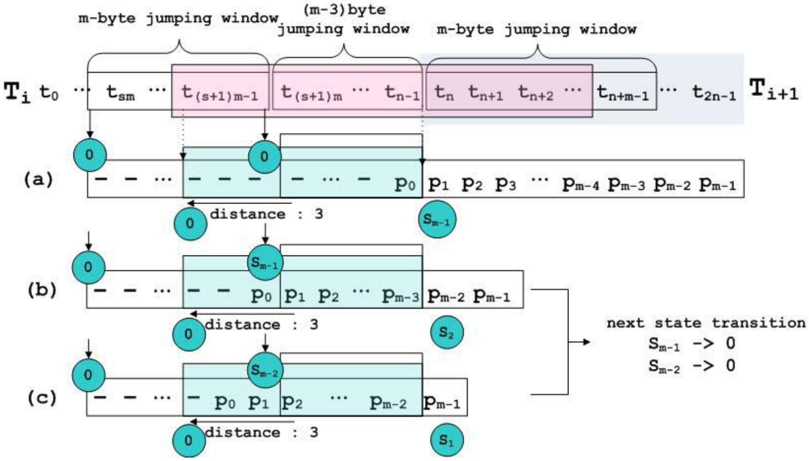

m-byte jumping window
(m-3)byte jumping window
m-byte jumping window
T_i t_0 ··· t_{sm} ··· $t_{(s+1)m-1}$ $t_{(s+1)m}$ ··· t_{n-1} t_n t_{n+1} t_{n+2} ··· t_{n+m-1} ··· t_{2n-1} T_{i+1}
(a)
0
0
– – ··· – – – – ··· – p_0 p_1 p_2 p_3 ··· p_{m-4} p_{m-3} p_{m-2} p_{m-1}
0
distance : 3
S_{m-1}
(b)
0
S_{m-1}
– – ··· – – p_0 p_1 p_2 ··· p_{m-3} p_{m-2} p_{m-1}
0
distance : 3
S_2
(c)
0
S_{m-2}
– – ··· – p_0 p_1 p_2 ··· p_{m-2} p_{m-1}
0
distance : 3
S_1
next state transition
S_{m-1} -> 0
S_{m-2} -> 0

Fig. 3. Example of multi-packet inspection processing: shifted state transition for alignment of a search window

For the split pattern matching, states can move to other states in order to align the last window if the previous partial match is done successfully. The state transition diagram should have this state transition information for the alignment. With the extended state transition, split pattern into the next packet payload can be easily matched. Furthermore, it requires storing only the last state information instead of the large packet reassembly buffer.

4 Performance Evaluation

For the evaluation of the multi-gigabit rate pattern-matching in NIDS, we have implemented IDS microblock which is a microcode program of Intel IXP28XX NP [10] to detect intrusion patterns in the packet payload. The IXP28XX NP development platform consists of dual network processor units (NPUs), 9-Mbit IDT's TCAM [11], and 10 ports of gigabit Ethernet (GbE). In this paper, we have generated packets matched with the Snort rule header using the traffic generator Smartbits 6000B. In the following experiments, throughputs are measured for various packet lengths with a single microengine (ME). Although the Intel IXP28XX NPU has 16 MEs, only 12 MEs are used to receive and transmit packets through external interfaces and process them in the current Intel's 10-port GbE IPv4/IPv6 forwarding application. We could add at most 4 MEs without modification of the current application for implementing our algorithm.

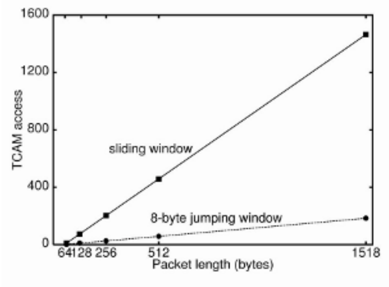


Fig. 4. Compares of TCAM Access

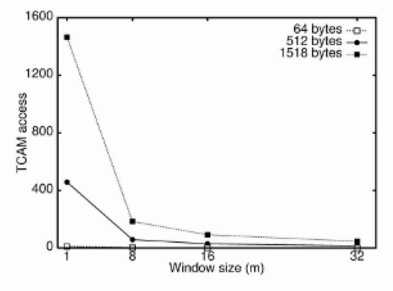


Fig. 5. Effects of the window size, *m*

Fig. 4 shows the result of TCAM access for pattern matching with varying the packet size for sliding window and 8-byte jumping window. For this experiment, only one ME is used for deep packet inspection among 16MEs. The number of TCAM access of sliding window increases rapidly as the packet length increases, while that of our algorithm increases slowly. With the maximum packet size, i.e., 1518 bytes in the Internet, the throughput of our proposed algorithm is about 1Gbps, while sliding window shows only the performance of about 200Mbps[12]. In this experiment, we proved that the number of TCAM access in the 8-byte jumping window is approximately 1/8 of the number of TCAM access in the sliding window scheme. Fig. 5 shows the effect of the window size, *m*. As the window size increases, the number of TCAM access is reduced 1/*m*.

5 Conclusion

In this paper, we have presented a multi-gigabit pattern-matching algorithm for network intrusion detection system in the high-speed network. The TCAM-based deep packet inspection algorithm developed in this paper uses a jumping window scheme, which is supported by position-aware sub-patterns and the state transition diagram to reduce the number of TCAM lookups. We have implemented the proposed algorithm on the Intel IXDP28xx platform. The performance of packet processing with our proposed algorithm is more than 3Gbps at the worst-case situation with the maximum packet size. We expect an increase of the performance through microcode optimization and window size augment. We've proven the feasibility of the proposed algorithm with our experimental implementation that runs on the IXDP28xx platform. In order to detect malicious attack split in two continuous packet, we extended the state transition diagram with shifting distance. Therefore, first pattern-matching in next packet is achieved easily with the state transition of previous packet. In this paper, we describe that the pattern length is the same value as window size, *m*. However, our proposed algorithm is applicable even if it is greater or lesser than window size *m*.

References

1. S. Dharmapurikar, P. Krishnamurthy, T. S. Sproull and J. W. Lockwood: Deep Packet Inspection using Parallel Bloom Filters in IEEE Micro, Vol. 24, No. 1, Jan. 2004, 52-61.
2. J. Lockwood, J. Moscola, M. Kulig, D. Reddick, and T. Brooks: Internet Worm and Virus Protection in Dynamically Reconfigurable Hardware in Military and Aerospace Programmable Logic Device (MAPLD), Sep. 2003.
3. I. Sourdis and D. Pnevmatikatos: Fast, Large-Scale String Match for a 10Gbps FPGA-based Network Intrusion Detection System inConference on Field Programmable Logic and Applications, Sep. 2003.
4. P. Jungck and S. S.Y. Shim: Issues in high-speed internet security in IEEE Computer Magazine, Vol. 37, No. 7, July 2004, 22-28.
5. M. Fisk and G. Varghese: Fast content-based packet handling for intrusion detection in Tech. Report CS2001-0670, UCSD, May 2001.
6. S. Wu and U. Manber: A fast algorithm for multi-pattern searching in Tech. Report, TR94-17, University of Arizona, May 1994.
7. J. Bo and L. Bin: High-speed discrete content Sensitive pattern match algorithm for deep packet filtering in Int'l Conf on Computer Networks and Mobile Computing, 2003.
8. F. Yu, R. H. Katz and T. V. Lakshman: Gigabit rate packet pattern-matching using TCAM in IEEE Int'l Conf on Network Protocols, Oct. 2004, 174-183.
9. J. Sung, S. Kang, Y. Lee, T. Kwon, and B. Kim: A Multi-gigabit Rate Deep Packet Inspection Algorithm using TCAM in IEEE Globecom, Nov. 2005.
10. Intel: Intel 2800 Network Processor in Hardware Reference Manual, Jan. 2004.
11. IDT: Integrated IP Co-Processor (IIPC) with QDR Interface in IDT75K52134/ IDT75K62134 User Manual, Sep. 2002.
12. S. Kang, I. Song, Y. Lee, and T. Kwon: Design and Implementation of a Multi-gigabit Intrusion and Virus/Worm Detection System in IEEE ICC, June 2006 (to appear).

Multicast OLSP Establishment Scheme in OVPN over IP/GMPLS over DWDM

Jeong-Mi Kim[1], Oh-Han Kang[2], Jae-Il Jung[3], and Sung-Un Kim[1,⋆]

[1]Pukyong National University, 599-1 Daeyeon 3-Dong Nam-Gu, Busan,
608-737, Korea
kimjm@pknu.ac.kr, kimsu@pknu.ac.kr

[2]Andong National University, 388 Song-chon Dong, Andong, Kyoungbuk,
760-749, Korea
ohkang@andong.ac.kr

[3]Hanyang University, 17 Haengdang-Dong Seongdong-Gu, Seoul, 133-791, Korea
jijung@hanyang.ac.kr

Abstract. OVPN (Optical Virtual Private Network) over IP (Internet Protocol)/GMPLS (Generalized Multi-Protocol Label Switching) over DWDM (Dense Wavelength Division Multiplexing) technology with QoS assurances is considered as a promising approach for the next generation OVPN. In this paper, we suggest a multicast OLSP (Optical Label Switched Path) establishment mechanism for supporting high bandwidth multicast services. For the establishment of the multicast OLSP, we propose a new multicast tree generation algorithm VS-MIMR (Virtual Source-based Minimum Interference Multicast Routing) that finds the minimum interference path between virtual source nodes. We also suggest an entire OVPN control mechanism to adapt the operation of the routing and signaling protocols of GMPLS.

1 Introduction

OVPNs are expected to be one of the major applications in the future optical networks. Therefore the OVPN over IP/GMPLS over DWDM technology has been suggested as a favorable approach for realizing the next generation VPN services[1].

In this paper, the characteristics of the OVPN multicast services are analyzed. And the establishment of multicast OLSP is investigated in two steps; OLSP establishment preparation phase and OLSP establishment phase. In the OLSP establishment phase, a new multicast tree generation algorithm, VS-MIMR is proposed. We also suggest an entire OVPN control mechanism to establish multicast OLSP adapting the operation of the routing and signaling protocols of GMPLS.

The rest of this paper is organized as follows. In section 2, we describe the functional architecture and operation of the QoS guaranteed OVPN. In Sections 3, we propose the multicast OLSP establishment scheme. In Section 4, the conclusion is presented.

[⋆] Corresponding author.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1163–1168, 2006.

2 Functional Architecture and Operation of QoS Guaranteed OVPN

We propose the functional architecture for providing the optical QoS in OVPN as shown in Fig.1. For establishing the multicast OLSP, a Customer Agent requests a CE (Client Edge)-to-CE OLSP establishment with SLA(Service Level Agreement) parameters to the Negotiation Policy Agent. Once the Negotiation Policy Agent in an ingress PE (Provider Edge) receives a trigger for setting up an OLSP, it invokes the QoS Routing Policy Agent for routing and wavelength assignment with the QoS parameters extracted from the request.

Based on the OVPN membership and resource information gathered by OSPF-TE+ (Open Shortest Path First with Traffic Engineering extensions)[2] and MP-BGP (Multi-Protocol Border Gateway Protocol)[3], the OVPN Routing Agent calculates the QoS guaranteed tree for establishing the multicast OLSP. In this process, the VS-MIMR algorithm choosing a tree is proposed to calculate the QoS guaranteed multicast tree. After the tree calculation, the OVPN Signaling Agent in the control plane is invoked to reserve the optical resource with the GMPLS signaling protocol, the RSVP-TE+ (Resource ReSerVation Protocol with Traffic Engineering extensions)[4].

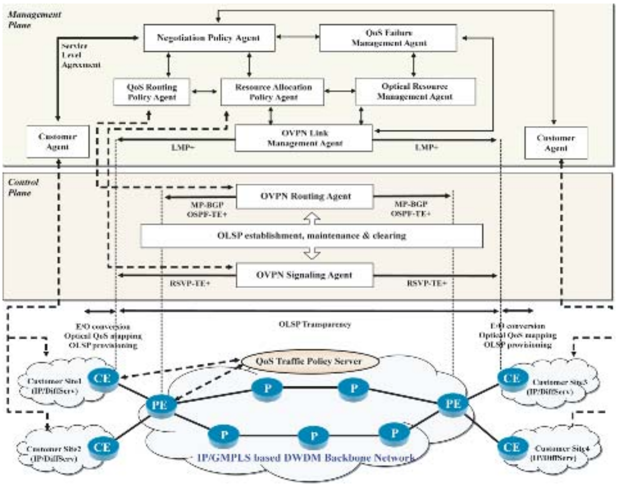

Management Plane
Negotiation Policy Agent
QoS Failure Management Agent
Service Level Agreement
QoS Routing Policy Agent
Resource Allocation Policy Agent
Optical Resource Management Agent
Customer Agent
OVPN Link Management Agent
LMP+
LMP+
Customer Agent
Control Plane
OVPN Routing Agent
MP-BGP OSPF-TE+
MP-BGP OSPF-TE+
OLSP establishment, maintenance & clearing
OVPN Signaling Agent
RSVP-TE+
RSVP-TE+
E/O conversion Optical QoS mapping OLSP provisioning
OLSP Transparency
E/O conversion Optical QoS mapping OLSP provisioning
QoS Traffic Policy Server
Customer Site1 (IP/DiffServ)
CE
PE
P
P
P
P
P
PE
CE
Customer Site3 (IP/DiffServ)
Customer Site2 (IP/DiffServ)
CE
CE
Customer Site4 (IP/DiffServ)
IP/GMPLS based DWDM Backbone Network

Fig. 1. The functional architecture of QoS guaranteed OVPN

3 Multicast OLSP Establishment

3.1 Preparation Mechanism for OLSP Establishment

Establishment of CE-to-CE Control Channel by LMP (Link Management Protocol): The control channels are used to exchange the control plane information such as the link provisioning, fault management information, label distribution information (implemented using a signaling protocol such as RSVP-TE+), and network topology, state distribution information (implemented using traffic engineering routing protocols such as OSPF-TE+).

The two core procedures of the LMP are the control channel management and link property correlation[5]. The control channel management is used to establish and maintain the control channels between the adjacent nodes. This is done by using the CONFIG and HELLO messages exchange. The link property correlation is used to synchronize the TE link properties and verify the TE link configuration.

***Routing Information Exchange by OSPF-TE+*:** In this paper, we assume the routing information is distributed by OSPF-TE+[2] between the PE nodes. The connection with the adjacent nodes is established by exchanging the Hello packet between the adjacent nodes. And then, only the LSA (Link State Advertisements) headers are exchanged and the recent needed information among them is checked through the Database Description packet. In the procedure of the database exchange, the recent needed information is requested through the Link State Request packet and the Link State Update packet containing the LSAs (Router-LSA, TE-LSA, and etc.) transmits the routing information.

***Routing Information Exchange by MP-BGP*:** The MP-BGP is an extended BGP-4 protocol for the exchange of not only the IPv4 routing information but also the routing information of the diverse network layer protocols[3]. It is also used for the exchange of the membership information among the customer sites in the same OVPN. In the procedure of the neighbor connection, the adjacent relation is set with other nodes by using the OPEN message, and the negotiation of the related parameters (autonomous system number, version of BGP, BGP Router ID, and etc.) are exchanged. The routing information is exchanged by using a UPDATE message.

After forming such an entire routing table of the OVPN, the QoS guaranteed path is established through the SLA negotiation procedure at the time of a connection request. The appropriate OLSP is calculated by the mechanism explained in the next section.

3.2 QoS Guaranteed Tree Establishment Mechanism

***SLA Negotiation for Multicast Session*:** In the case of the multicast OLSP establishment, the SLA negotiation procedure is also required between the OVPN backbone network and the customer site to establish the QoS guaranteed multicast tree.

When the QoS-TP (Traffic Policy) server (we assume this server is belong to the management plane in Fig.1) receives the SLA request that contains the multicast session information and the QoS parameters, it sends the VS_QUERY message to all the VSs in the OVPN network so that the PVS (Primary Virtual Source) and SVSs (Secondary Virtual Sources) can be found as shown in Fig.2. All the VSs respond to the QoS TP server with the VS_REPORT message.

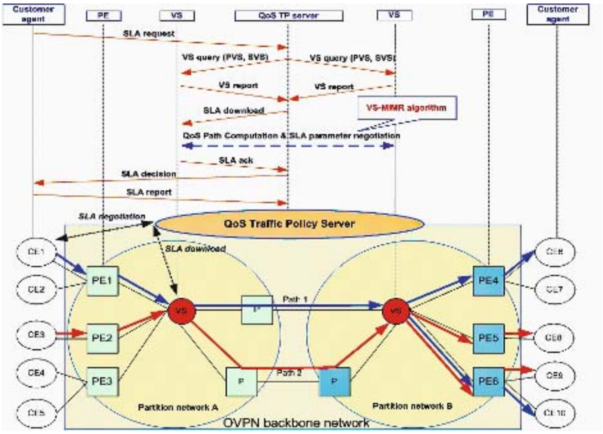


Fig. 2. SLA negotiation procedure for multicast service

When the QoS-TP server gets the information of all PVS and SVSs, it downloads the SLA parameters onto the Negotiation Policy Agent of the PVS in order to establish the connections to all the SVS nodes. At this time, using the VS-MIMR algorithm, we improve the resource utilization in the OVPN backbone network.

***VS-MIMR for Multicast Tree Generation*:** The VS-MIMR algorithm is suggested to choose minimum interference paths. The proposed algorithm overcomes the limitation of VS-based method[6] and provides an efficient use of wavelengths.

Fig.2 illustrates the VS-MIMR algorithm. There are two potential source-destinations pairs such as (PE1, PE4&PE6) and (PE2, PE5&PE6). When Path 1 is chosen for the first multicast session in order to make a resource reservation for the path between a PVS-SVS pair, the other multicast session may share the same path having a minimum-hop path. It can lead high blocking probability by inefficiently using the resource due to the traffic concentration on that path. Thus, it is better to pick Path 2 that has a minimum interference effect for other future multicast session requests even though the path is longer than Path 1. We define that a segment means a path between VS nodes. And each segment must follow the wavelength continuity constraint, because only VS nodes can have a wavelength splitting and conversion capability. We define some additional notations used in this algorithm as follows.

• $G(N, L, W)$: The given network, where N is the set of nodes, L is the set of links, and W is the set of wavelengths per link.
• (v_p, v_s): A PVS-SVS node pair.
• (a, b) : A PVS-SVS node pair for current requests, where $(a,b) \in (v_p,v_s)$.
• Λ: The set of potential PVS-SVS node pairs that can be requested by multicast session in the future, where $\forall(v_p,v_s) \in \Lambda$
• S^n_{ps}:The nth segment of the set of minimum hop segments connecting the path.
• C_{ps}: The set of critical links between the (v_p,v_s) pair, that is, C_{ps} are shared with other node pairs at the same time.

• F_{ps}: The number of available wavelengths on bottleneck segment that has the smallest residual wavelengths.
• $R(S^n_{ps})$: The number of residual wavelengths on the segment S^n_{ps}.
• α_{ps}: The weight for a segment according to the degree of multicast session resource reservation requests between the PVS node and SVS node.
• $w(S^n_{ps})$: The accumulated total weights for S^n_{ps}.
• $\mathit{\Delta}$: A threshold value of available wavelengths on S^n_{ps} (30% of the total wavelengths in S^n_{ps}).

Based on these notations, the segment weights are determined as follow:

$$Max \sum \alpha_{ps} \cdot F_{ps}. \tag{1}$$

$$CP_{ps}\colon (S^n_{ps}\colon e \in C_{ps}) \cap (R(S^n_{ps}) < \triangle). \tag{2}$$

$$w(S^n_{ps}) = \sum_{\forall (v_p, v_s) \in \mathit{\Lambda} \backslash (a,b)} \alpha_{ps}(\partial F_{ps}/\partial R(S^n_{ps})). \tag{3}$$

$$\{^{\partial F_{ps}/\partial R(S^n_{ps})=1[if(v_p,v_s):S^n_{ps}\in CP_{ps}]}_{\partial F_{ps}/\partial R(S^n_{ps})=0[otherwise]} \tag{4}$$

$$w(S^n_{ps}) = \sum_{(v_p,v_s):S^n_{ps}\in CP_{ps}} \alpha_{ps}. \tag{5}$$

Equation (1) represents the minimum interference of the wavelength path decision between the PVS node and SVS node. Equation (2) determines the *CP* (congestion path) with congestion possibility for potential future connection requests between the VS nodes.

We presents the weight of each segment for all (v_p, v_s)-pairs in the set $\mathit{\Lambda}$ except the current request when setting up a connection as shown in equation (3). And equation (4) allocates the differentiated values to the nth segment between the VS nodes. Calculating the weight of all nodes is difficult, so we apply equation (4) to equation (3). Finally, computing the segment weights is simplified as shown in equation (5). Therefore, the VS-MIMR decides a wavelength path that has a minimum value of segment weight $w(S^n_{ps})$.

3.3 Multicasting Distribution Tree Construction Using RSVP-TE+

After the tree calculation, a point-to-multipoint OLSP tree (P2MP tunnel) must be constructed by the RSVP-TE+ extensions for multicasting services. Although the P2MP OLSP is constituted of the multiple source-to-one leaf (S2L) sub-OLSPs, we can signal all S2L sub-OLSPs in one PATH message with the EXPLICIT_ROUTE object (ERO), P2MP SECONDARY_EXPLICIT_ROUTE object (SERO), and S2L_SUB_LSP object (S2LO)[7]. Fig.3 shows a P2MP OLSP with PE1 as a source node and three destination nodes (PE2, PE3 and PE4). When the branch nodes (VS1 and VS2) receive the PATH message, it generates the multiple PATH messages with the different EROs and SEROs. After sending out the PATH messages to all nodes of the multicasting tree, this will be confirmed by the RESV message with the ROUTE_RECORD object (RRO)

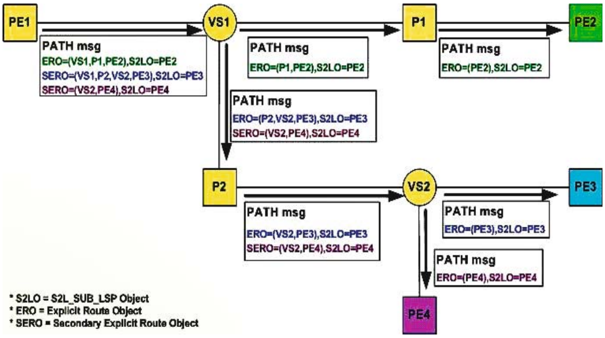


Fig. 3. Multicasting distribution tree construction

and P2MP SECONDARY_ROUTE_RECORD objects (SRROs) at each link of the multicasting tree[7].

4 Conclusion

In this paper, the functional architecture and the interoperation of the control plane and management plane of the QoS guaranteed OVPN are proposed. For the establishment of the multicast OLSP, we propose the VS-MIMR algorithm that finds the minimum interference path between virtual source nodes. We also suggest an entire OVPN control mechanism to adapt the operation of the routing and signaling protocols of GMPLS.

Acknowledgment

This work was supported by grant No.(R01-2003-000-10526-0) from Korea Science & Engineering Foundation.

References

1. Z. Zhang, et al., An Overview of Virtual Private Network (VPN): IP VPN and Optical VPN, Photonic Network Communications, vol.7, no.3, pp.213-225, 2004.
2. K. Kompella and Y. Rekhter, OSPF Extensions in support of Generalized Multi-Protocol Label Switching, IETF RFC 4203, Oct. 2005.
3. T. Bates et al, Multiprotocol Extension for BGP-4, RFC2858, June 2000.
4. L. Berger, GMPLS Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions, IETF RFC 3473, Jan. 2003.
5. J. Lang, Link Management Protocol (LMP), IETF RFC4204, Oct. 2005.
6. N. Sreenath et al., Virtual Source Based Multicast Routing in WDM Optical Networks, Photonic Network Communications, vol.3, no.3, pp.217-230, 2001.
7. R. Aggarwal et al., Extensions to RSVP-TE for Point to Multipoint TE LSPs, draft-ietf-mpls-rsvp-te-p2mp-01.txt, IETF Internet Draft, June 2005.

Directional Reception vs. Directional Transmission for Maximum Lifetime Multicast Delivery in Ad-Hoc Networks[⋆]

Kerry Wood and Luiz A. DaSilva

The Bradley Department of Electrical
and Computer Engineering,
Virginia Tech
woodk@vt.edu, ldasilva@vt.edu

Abstract. In this paper, we present a mixed-integer linear program (MILP) designed to optimize max-min path lifetime for multicasts in directional antenna equipped networks in the presence of interference. We then employ the MILP to perform a head-to-head comparison between directional transmission and directional reception. We also propose and analyze a new directional reception heuristic. Our results indicate that directional reception can match directional transmission when extending path lifetime, with lower complexity and employing routes that use much less cumulative power.

Keywords: directional antennas, optimization, maximum-lifetime, multicast.

1 Introduction

We investigate the construction of multicast trees that maximize path lifetime for ad-hoc networks where nodes are equipped with directional antennas. We compare the use of directional antennas for the *transmission* of signals (which we refer to as D-TX) versus their use for *reception* (which we refer to as D-RX). Specific contributions include:

1. To show that, while directional transmission can match the path lifetime obtained with directional reception, it does so at the expense of considerable increase in cumulative power and complexity;
2. To produce a mathematical formulation of the optimization problem that takes into account potential interference among links that are part of the same multicast tree; and
3. To propose a heuristic for forming a multicast tree using directional reception and to show that this heuristic outperforms previously proposed heuristic solutions that employ directional transmission.

[⋆] This work was partially supported by a National Science Foundation Integrated Graduate Education and Research Training (IGERT) grant (award DGE-9987586).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1169–1175, 2006.

Table 1. Notation

Symbol	Definition
$\mathcal{N}$	Set of network nodes.
$\mathcal{R}$	Set of nodes that are receivers.
$\mathcal{B}_i$	Set of beams available at node i.
$\mathcal{B}_i(j)$	Set of beams at node i where j is within main lobe.
s	Source node.
P_i^t	Transmission power level at node i
$P_{i,b}^t$	Transmission power, node i using beam b.
$\boldsymbol{G_i}$	Gain / path loss vector at node i.
$F_{i,j,b}$	1 if flow possible from i to j when i using beam b.
$F_{i,j}$	total flow indicator from node i to node j ("super-flow").
$M_{i,j}$	Message (information) flow from i to j.
$B_{i,b}(k)$	Beam gain from node i to node k, using beam b ($b \in \mathcal{B}_i$)
$U_{i,j,b}$	Bound of interference at j when receiving from i using beam b.
Q	Large integer (Big-M).
S^i	SINR ratio required at node i.
N_t	Thermal / ambient noise.
R_i	Energy remaining (battery) available at node i.
P_{max}	Maximum transmit power setting for nodes.
θ_{min}	Minimum beamwidth for a node.
Pct^{inbeam}	Percentage of power in main beam lobe.
D-TX	Directional transmit MILP model.
D-RX	Directional receive MILP model.

We first introduce our mathematical program and associated communication model assumptions. Then, we utilize the model to compare the performance of directional transmission and reception optima, and to characterize the performance of heuristics designed to approximate both methods.

2 Mathematical Program

The mathematical program presented in this work incorporates the effects of inter-node and side-lobe interference through a signal to interference and noise ratio (SINR) sufficiency requirement. We refer the reader to Table 1 for the notation we adopt throughout the paper. Antennas assume a common bulb-and-cone model shown in Figure 2(a), where Pct^{inbeam} represents the fraction of power in the main antenna lobe.

The basis for the model is the SINR sufficiency constraint, in contrast to previous MILP models that do *not* account for interference [1][2]. Put simply, a logical link from node i to node j is feasible whenever the ratio of the power received from the intended transmitter to that received from all other sources exceeds the receiver's SINR requirement, as represented in Inequality 1.

$$P_{i,b}^t \cdot B_{i,b}(j) - S^j \cdot \left[\sum_{\substack{k \in \mathcal{N} \\ k \neq i,j}} \sum_{l \in \mathcal{B}_k} P_{k,l}^t \cdot B_{k,l}(j) + N_t \right] \geq 0 \qquad (1)$$

Inequality 1 represents SINR for a single beam configuration. The mathematical program requires that all possible configurations be enumerated, and

min T

s.t.

$$\begin{aligned}
P^t_{i,b} \cdot B_{i,b}(j) - S^i \cdot \left[\sum_{\substack{k \in \mathcal{N} \\ k \neq i,j}} \sum_{l \in \mathcal{B}_k} P^t_{k,l} \cdot B_{k,l}(j) + N_t \right] - \frac{Q}{P^{max}_i} \cdot \sum_{\substack{k \in \mathcal{N} \\ k \neq i,j}} P_{i,m} \geq & \\
F_{i,j,b} \cdot [U_{i,j,b} + Q] - [U_{i,j,b} + Q] &: \forall i, j \in \mathcal{N}, \forall b \in \mathcal{B}_i, j \neq s, i \neq j \\
F_{i,j} = \sum_{\forall b \in \mathcal{B}_i} F_{i,j,b} &: \forall i, j \in \mathcal{N}, i \neq j \\
F_{i,j} + F_{j,i} \leq 1 &: \forall i, j \in \mathcal{N}, i \neq j \\
\sum_{\substack{\forall k \in \mathcal{N} \\ k \neq j}} F_{k,j} \geq 1 &: \forall j \in \mathcal{R} \\
M_{i,j} \leq |\mathcal{R}| \cdot F_{i,j} &: \forall i, j \in \mathcal{N}, i \neq j \\
M_{i,j} \leq \sum_{\substack{\forall k \in \mathcal{N} \\ k \neq i,j}} M_{k,i} - 1 \cdot (1 : i \in \mathcal{R}) &: \forall i, j \in \mathcal{N}, i \neq j \\
P^t_i \geq P^t_{i,b} &: \forall i \in \mathcal{N}, \forall b \in \mathcal{B}_i \\
T \geq \frac{P^t_i}{R_i} &: \forall i \in \mathcal{N} \\
P^t_i \leq P_{max} &: \forall i \in \mathcal{N} \\
F_{i,j,b} \in \{0, 1\} &: \forall i, j \in \mathcal{N}, i \neq j, j \neq s, b \in \mathcal{B}_i
\end{aligned}$$

Fig. 1. D-TX: Formulation for Max-Min Network Lifetime with Inter-Node Interference

consequently, only a small subset will be satisfied for any given multicast configuration.

"Big-M" notation is introduced in Inequality 2, where $U_{i,j,b}$ relaxes the constraint unless the link is "active" as indicated by the binary variable $F_{i,j,b}$.

$$P^t_{i,b} \cdot B_{i,b}(j) - S^i \cdot \left[\sum_{\substack{k \in \mathcal{N} \\ k \neq i,j}} \sum_{l \in \mathcal{B}_k} P^t_{k,l} \cdot B_{k,l}(j) + N_t \right] \geq F_{i,j,b} \cdot U_{i,j,b} - U_{i,j,b} \quad (2)$$

With D-RX, a node seeks to maximize its SINR by orienting its antenna toward a single transmitter (the node's parent in the multicast tree). With D-TX, however, it may be desirable for a node in the tree to select an antenna beam that covers multiple receivers (the node's descendants in the multicast tree). Accordingly, the formulation of the D-TX optimization problem is more complex than that of D-RX, to account for the combinatorics of beam selection.

Our choice of max-min path lifetime ($\frac{1}{T}$) as our optimization metric reflects the metric used in previous work such as [3][4]. The time until death of the *first* node in a forwarding tree is defined as path lifetime. Wieselthier et al. showed that cumulative power (i.e. min-power) metrics do not correlate well to this metric [5]. Later, we show that our results are consistent with these findings.

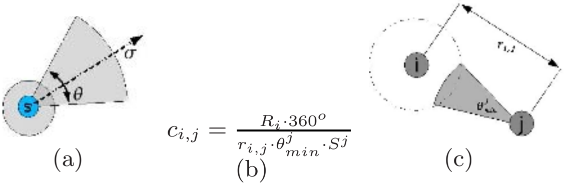
Table 2. Size of MILP Formulations

	D-TX:		D-RX:	
Type	Maximum Number	Contributor	Maximum Number	Contributor
Constraints	$O(\vert\mathcal{N}\vert^4)$	SINR	$O((2 \cdot \vert\mathcal{N}\vert)^2)$	SINR
Binary Var.	$O(\vert\mathcal{N}\vert^4)$	$F_{i,j,b}$	$O((2 \cdot \vert\mathcal{N}\vert)^2)$	$F_{i,j,b}$
Continuous Var.	$O(\vert\mathcal{N}\vert^3)$	$P_{i,j,b}$	$O(\vert\mathcal{N}\vert^2)$	$M_{i,j}$

Figure 1 shows the complete D-TX model as translated into the mixed-integer linear program (MILP) notation in Table 1. We refer the reader to [6] for a complete discussion of the D-RX model. From top to bottom, represented are: SINR constraints, "super-flow" variables ($F_{i,j}$), flow consistency inequalities, receiver demand, SINR message throttling ($M_{i,j}$), message consistency, "super-power" ($P_{i,j}$), path lifetime ($\frac{1}{T}$), power cap, and binary constraints. MILP programs are well known to be NP-hard [7], and empirically, the difficulty required to find a solution is often related to the number of binary variables in the model. Bounds on number of constraints and variables for the D-TX and D-RX mathematical programs are shown in Table 2. This confirms our previous observation that the formulation of the D-TX problem is considerably more complex than that of D-RX.

3 D-RX Heuristic

In this section, we introduce the Directional Reception Incremental Protocol (DRIP), a low-computation heuristic designed to approximate the MILP optima. The network is modeled as a directed graph $\mathcal{G}$ where an $i \rightarrow j$ link $l_{i,j}$ has a weight $c_{i,j}$ assigned as in Figure 2(b). Link weight is proportional to the amount of battery power remaining at the transmitting node, denoted by R_i, and the power required for inter-node communication. In this case, higher weight indicates longer lifetime. All network links are represented by $\mathcal{L}$, where $\mathcal{L}(i)$ denotes links with node i as a *destination.* A forwarding tree $\mathcal{T}$ is built from source to all receivers incrementally, until all receivers are included in the tree. At each step, the highest weight link available is added. Once the algorithm terminates, post-processing eliminates any branches that do not contain receivers. The heuristic is described in pseudocode in Algorithm 1.



(a)

$$c_{i,j} = \frac{R_i \cdot 360^o}{r_{i,j} \cdot \theta^j_{min} \cdot S^j}$$

(b)

(c)

Fig. 2. DRIP Link Cost

Algorithm 1. DRIP Multicast Routing Algorithm

```
$\mathcal{T} \leftarrow s$
$\mathcal{N} \leftarrow \mathcal{N} \setminus \{s\}$
$\mathcal{L} \leftarrow \mathcal{L} \setminus \mathcal{L}(s)$
while ! $\mathcal{R} \subseteq \mathcal{T}$ do
    $i \leftarrow$ highestAvailableLinkWeight($\mathcal{L}$)
    $\mathcal{T} += i$
    $\mathcal{L} \leftarrow \mathcal{L} \setminus \mathcal{L}(i)$
end while
removeUnNeededBranches($\mathcal{T}$)
```

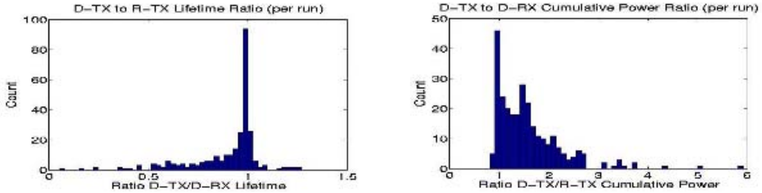
4 Results: D-TX vs. D-RX MILP Models

In this section, we illustrate the results of our head-to-head comparison of antenna use. The results are produced using our MILP models, and reflect the optimal path lifetime for the given network configuration and multicast group.

Experimental Setup
To perform a fair comparison, both the D-RX and the D-TX models were applied to identical networks. Because of the complexity of the D-TX model, network size is restricted to 10 nodes ($|\mathcal{N}| = 10$). We use the following parameters for all nodes $i \in \mathcal{N}$: $Pct^{inbeam} = 0.7$ (30% of energy lost to sidelobes), $\theta_{min} = 45^o$, $S^i = N_{thermal} = 1$, $P_{max} = 100$, $\alpha = 2$, $R_i = 300$. Nodes are randomly placed in 5×5, 10×10 and 15×15 unit areas. Receiver set size varies among five values $|\mathcal{R}| \in \{1, 3, 5, 7, 9\}$. For each combination of network dimension and receiver set size, 20 individual runs were performed, for a total of 300 runs. Mixed-integer linear programs are solved with either GLPK [8], or CPLEX [9].

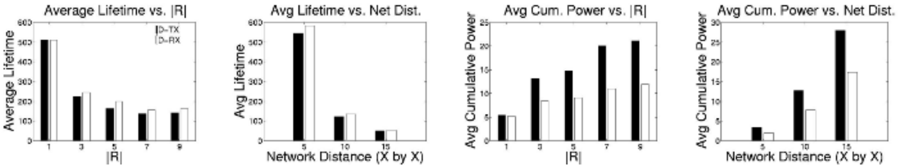
D-TX to D-RX Comparison
Figure 3 displays histograms of both max-min lifetime, and cumulative power use ratios over all runs. As illustrated in Figure 3(a), in most cases D-RX produces identical or superior max-min lifetime values to those of D-TX. More importantly, Figure 3(b) clearly indicates that D-RX achieves this with *much* less overall power. For a significant number of test networks, D-TX requires over 200% of the power of D-RX.

Figure 4 dispenses with the histogram illustration, and shows actual lifetime ((a),(b)), and cumulative power values ((c),(d)) against a single network



(a) $\frac{D-TX}{D-RX}$ Lifetime (b) $\frac{D-TX}{D-RX}$ Cumulative Pow

Fig. 3. Performance Histograms (Optima)



(a) (b) (c) (d)

Fig. 4. Ratios vs. Single Network Parameter (Optima)

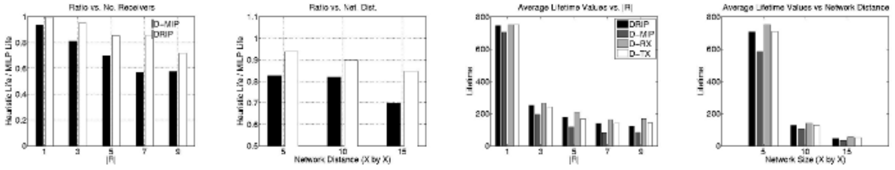
parameter. The weak correlation between path lifetime and cumulative power is evident. Figures 4(a) and (b) clearly show that for any breakout by $|\mathcal{R}|$ or network distance, D-RX and D-TX achieve comparable path lifetime. Figures 4(c) and (d), however, clearly show that D-RX requires significantly less cumulative power.

5 Analysis of Heuristic Methods

This section builds upon the analysis of the D-TX and D-RX MILP models. Here, we compare a previously defined directional *transmission* heuristic D-MIP [3] and our own directional *reception* heuristic DRIP. These heuristics are compared head-to-head, and also to their respective MILP optima.

Experimental Results
Figure 5(a) and (b) show the average path lifetime obtained by DRIP and D-MIP heuristics, always normalized to the optimum path lifetime returned by the MILP. In all graphs, the ratio shown is the heuristic compared to the MILP optimal value. For example, for $|\mathcal{R}| = 9$, D-MIP only achieves approximately 55% of the MILP optimal, while DRIP achieves over 70%.

Clearly, the MILP model provides an upper bound on path lifetime for either heuristic under the effects of interference. While both methods suffer with larger receiver sets and bigger network distances, D-MIP's performance deteriorates much faster with both. As the size of the receiver set increases, D-MIP's ability to approximate the optimal value declines. Recall, also, that D-RX can also have a higher lifetime value, meaning that the approximation, *and* the target value



(a) (b) (c) (d)

Fig. 5. Heuristic Performance Ratios

are larger. Figures 5(b) and (c) show the actual average lifetime values for the schemes in question.

6 Conclusion

This paper investigates the use of directional antennas for multicast delivery in ad-hoc networks. Our main contribution is to show that, while directional transmission can match directional reception in terms of max-min path lifetime, it does so at the expense of considerably higher power expenditure and complexity.

We present a mixed-integer linear program for finding the optimal antenna configuration, power settings, and logical topology for max-min path lifetime under the effects of *interference* when using directional antennas. The existing literature has focused on the use of antennas for directional *transmission*, and ignored interference. Our results (under more realistic assumptions) provide evidence that nodes are better served using the antennas for *reception*. Results from a simple heuristic for multicast tree selection employing directional reception further confirm these findings.

References

1. S. Guo and O. Yang, "Antenna orentiation optimization for minimum-energy multicast tree construction in wireless ad hoc networks with directional antennas," in *Proc. of MobiHoc*, pp. 234–243, May 2004.
2. S. Guo and O. Yang, "Minimum energy multicast routing for wireless ad-hoc networks with adaptive antennas," in *Proc. of International Conference on Network Protocols (ICNP)*, pp. 151–160, October 2004.
3. J. Wieselthier, G. Nguyen, and A. Ephremides, "On the construction fo energy-efficient broadcast and mutlicast tree in wireless networks," in *Proc. of IEEE INFOCOM*, 2000.
4. J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proc. of IEEE INFOCOM*, 2000.
5. J. Wieselthier, G. D. Nguyen, and A. Ephremides, "Algorithms for energy-efficient multicasting in static ad-hoc wireless networks," in *ACM Mobile Networks and Applications (MONET)*, pp. 6(3) 251–263, June 2001.
6. K. N. Wood and L. A. DaSilva, "Optimization of network lifetime with directional listening," in *Proc. of IEEE BROADNETS*, October 2005.
7. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness.* NY: W.H. Freeman and Company, 1979.
8. http://www.gnu.org/software/glpk/glpk.html.
9. http://www.cplex.com/.

Micro- and Macroscopic Analysis of RTT Variability in GPRS and UMTS Networks

Jorma Kilpi[1,⋆] and Pasi Lassila[2]

[1] VTT Information Technology, P.O. Box 12022, FIN 02044 VTT, Finland
Jorma.Kilpi@vtt.fi
[2] Helsinki University of Technology, P.O. Box 3000, FIN 02015 HUT, Finland
Pasi.Lassila@hut.fi

Abstract. We study the data from a passive TCP/IP traffic measurement from a Finnish operator's GPRS/UMTS network. Of specific interest is the variability of Round Trip Times (RTTs) of TCP flows. The RTTs are analysed at micro- and macroscopic level. The microscopic level involves detailed analysis of the RTTs of individual flows, and we are able to detect, e.g., periodic behavior (via Lomb periodogram) and rate changes in the radio channel. At the macroscopic level we focus on the impact of so called self-congestion caused by bandwidth sharing at the mobile device itself, and it is shown how this seriously affects the RTTs observed by a given flow, both in GPRS and in UMTS.

Keywords: traffic measurements, RTT variability, GPRS, UMTS.

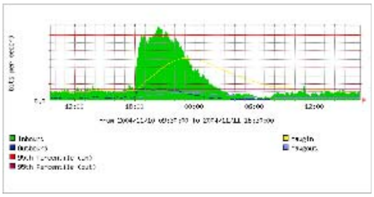
1 Introduction

We study the data from a passive traffic capture measurement representing a 30 hour TCP/IP trace measured from one GGSN node of a GPRS/UMTS network in a major Finnish operator's network. The objective of the measurement is to analyze the variability of the Round Trip Times (RTTs) of TCP flows, where one end point of the flow is in the GPRS/UMTS network and the other in the public Internet. Large variability of the RTTs may cause problems to TCP's retransmission methods causing spurious timeouts (see, e.g., [1]). We perform a detailed analysis of the RTTs of selected individual flows (microscopic analysis) and the aggregate traffic (macroscopic analysis). Furthermore, as the measurement data contains TCP flows both from GPRS users and UMTS users, we are able to compare the properties of the RTT process for both technologies.

Some measurements of RTT variation have been recently made in mobile networks see, e.g., [1], [2], and [3]. Notably, in [1] an algorithm is provided for detecting spurious events in TCP, and [3] analyses RTTs, loss and throughput characteristics, among other things. However, our focus is different than in these studies. We study how individual flows observe the RTTs and the aggregate RTTs of all flows. We also experiment with other potentially useful statistical tools not used in earlier studies, namely Lomb periodograms (LP) as motivated by [4] and wavelets. However, the results on wavelets are not in this paper due to space limitations, but can be found in [5]. Finally, we discuss in

[⋆] The authors thank Vesa Antervo from Elisa for his efforts in obtaining the trace and Marco Mellia for his help on tstat.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1176–1181, 2006.



Sample Number	TCP Connection Arrival Time	Flat Rate Population Present?
1	10:30-18:00	No
2	18:00-21:00	Yes
3	21:00-24:00	Yes
4	00:00-06:00	Yes
5	06:00-16:30	No

Fig. 1. Impact of flat rate population on traffic profile (left) and division into subsamples according to TCP connection arrival time (right)

the macroscopic level analysis the clear impact of simultaneous TCP connections (self congestion) on RTTs, which has not been addressed in previous studies.

2 Measurement Setup and Classification Methodology

The measurement point was the monitoring port of the Gi interface of one of the GGSN elements. The measurement was planned such that a statistically representative sample of the traffic was obtained. We then verified the accuracy of the time stamps in the data - the accuracy proved to be $\pm 50 \mu s$ (i.e., more than sufficient). The flow-level statistics were obtained by using `tstat`[1]. Additionally, `tstat` was modified to record all valid RTT samples of the TCP flows.

Traffic profile: The trace was obtained from a 30 hour measurement on Nov 10, 2004 and consists of traffic from GPRS and piloting UMTS users. Some of the subscribers have only volume based charging but some portion of the subscribers had also flat rate between 18.00-06.00. The effect of this *flat rate population* is significant, see Figure 1 (left). The data has been divided into 5 groups, as shown in Figure 1 (right), according to the TCP connection *arrival time*, which is determined by the time stamp of the SYN packet sent by the client. To study the impact of tariff change, we compare the data from Sample 1 and 2.

Semi-RTT and RTT count: We focus on the properties of RTTs as experienced by TCP flows. As we are not measuring directly at the sender/receiver, the RTT process can not be fully observed. Instead, the notion of semi-RTT is used, similarly as in [2]. Semi-RTT refers to the difference between the time stamps of a (downstream) TCP/IP packet carrying data payload and of the corresponding ACK packet. The size of a flow is measured in terms of the field *RTT Count*, and it is one of the parameters `tstat` provides for each completely observed TCP flow. RTT Count represents the number of times a data segment and the corresponding ACK has been observed, and it is provided both for the upstream and downstream connections.

GPRS vs. UMTS flows: During the measurement there were only piloting UMTS users, but the absolute number of observed TCP connections that could be associated to UMTS was sufficient to make comparative analysis against GPRS. Due to the measurement setup, exact identification of whether a flow originated from GPRS or UMTS

[1] `http://tstat.tlc.polito.it`

was not possible. However, RTTs in UMTS are about one magnitude smaller than in GPRS [2]. Thus, flows with a minimum RTT less than a given threshold were identified as UMTS flows. The threshold value we used was 0.4 ms (in [2] a minimum of 0.476 ms was observed for GPRS).

3 Microscopic Analysis of Long TCP Flows

Some specific flows are analyzed first to give the reader a flavor of what is behind the macroscopic level analysis. Example flows were chosen because the time series $\{(t_i, RTT(t_i)) \mid i = 1, \ldots, \text{RTT Count}\}$ had some distinctive features, where t_i is the time stamp of an ACK packet and $RTT(t_i)$ is the RTT value calculated from the ACK packet. Additionally, we focused only on the very longest flows to be able to detect clear changes over a comparably long time interval.

Flow 1 (GPRS): The first flow was chosen since we wanted to understand what caused the improvement in RTT values after about half an hour as shown in Figure 2 (left), and we wanted to see if the rise in traffic due to tariff change at 18:00 affected this flow. Improvement in RTTs after about half an hour was easily seen to be due to change from large (1380B) to a smaller (536B) segment size. Analysis of the receiving rate of packets revealed that the source is receiving data at a constant rate of 18.2 kbit/s. More detailed inspection showed that a data burst of fixed size was sent every 3rd second on the average. It also takes about 3 seconds before every segment of one burst has been acknowledged by the mobile host. After the change in the segment size, the data burst could be sent slightly better within the 3 second interval. Thus, Flow 1 is probably produced by a streaming audio application, though it was behind TCP port 80. The application had a 3 second play-out buffer, and the application forced the change in the segment size. Finally, another thing to notice was that there were no simultaneous TCP connections from the same mobile.

To further analyse the periodic behavior, the LP has been computed corresponding to the beginning, middle and end of the flow, see Figure 2 (middle and right). From the middle plot it is clear that in the beginning the burst period is slightly above 3 seconds. After the change in the segment size, the period is somewhat below 3 seconds. Examining Figure 2 (left) one can observe that after the tariff change at 18:00 the level of the RTTs rises slightly again (possibly due to increased traffic in the network). Computing the LP from the end shows that the burst period increases to slightly above 3 seconds again, see Figure 2 (right).

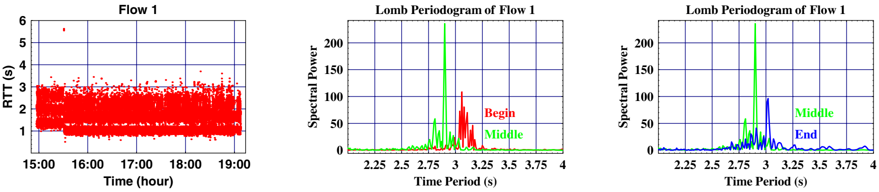


Fig. 2. Time series of Flow 1 (left) and its LPs (middle and right)

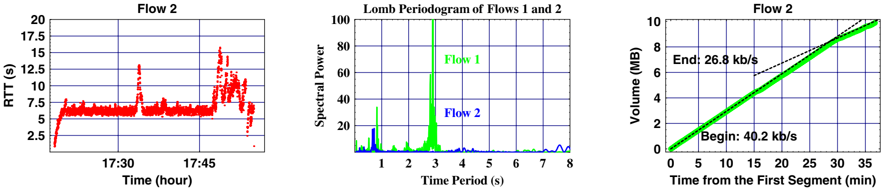


Fig. 3. Time series of Flow 2 (left) and its LP (right)

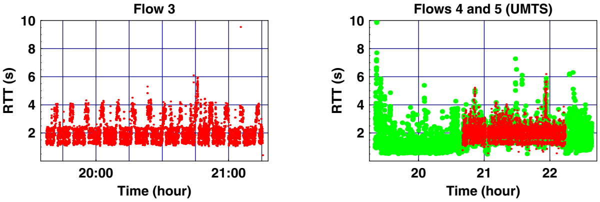


Fig. 4. Time series of Flow 3 (left) and Flows 4 and 5 (right)

Flow 2 (GPRS): As seen in Figure 3 (left) RTTs at a level of 5.0-7.5 seconds seems to be normal for this connection. Unlike for Flow 1, the use of LP revealed no significant periodic structure for Flow 2, see Figure 3 (middle). There it can be seen that Flow 2 has a small spike in the LP at slightly below 1 second, but it is negligible compared to the pronounced spike of Flow 1 at 3 seconds. Moreover, as Figure 3 (right) indicates, there was a change in the downlink capacity from 40.2 kb/s to 26.8 kb/s, which corresponds to a loss of one downlink PDCH when using CS-2 channel coding scheme. Analysis revealed that this was due to self-congestion caused by other simultaneous TCP connections from the same mobile host. The high overall level of RTTs is probably due to a low terminal capacity.

Flow 3 (GPRS): Again, use of LP showed no regular periodicity. However, Flow 3 had some peculiar regular intervals of bad RTTs. In this case the self-congestion is also an explanation since the user was simultaneously running an application (MSNP) which initiated a new flow in port 80 regularly with approximately 8 minute intervals downloading a file of size 265.3 kB. These downloads occur exactly at the bad intervals visible in Figure 4 (left).

Flows 4 and 5 (UMTS): Both flows originated from the same mobile, and they are shown in Figure 4 (right). The level of RTTs of Flow 4 increase when Flow 5 starts. There were also several other long flows simultaneously from the same mobile. Almost all RTT variations are explained by these other flows.

4 Properties of RTT at the Macroscopic Level

Changes in quantiles: We computed the empirical CDFs from the aggregate RTT data. Figure 5 (left) compares Samples 1 and 2 and we can see a clear shift in the quantiles. Based on packet level data analysis (results not shown here due to lack of space), we

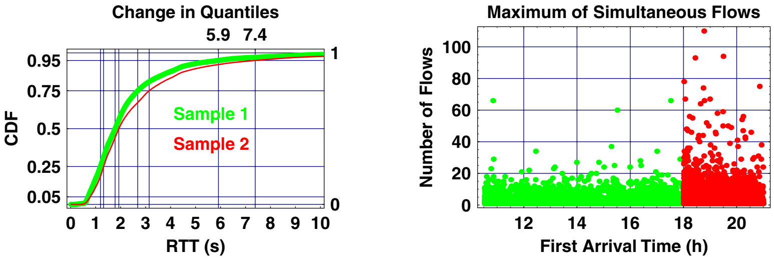


Fig. 5. Comparison of CCDFs (left) and increased web activity after 18:00 (right)

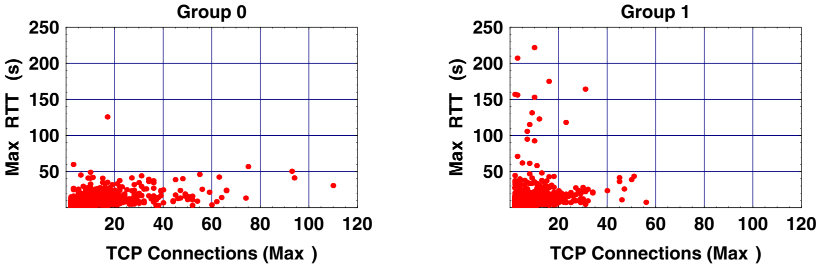


Fig. 6. Max RTTs with no upstream traffic (left) and with upstream traffic (right)

argue that the network is not congested but the shifts in the quantiles are due to self-congestion caused by an increased amount of simultaneous TCP connections from a given mobile (bandwidth sharing). Indeed, in Figure 5 (right) we show as a measure of web activity the maximum number of simultaneous TCP flows from a given mobile (IP address). Web activity clearly rises after 18:00 (corresponding to the data in Sample 2).

Self-congestion: To further study the effect of self-congestion on the RTTs, we divided such GPRS sessions, that had at least two simultaneous TCP connections, into two groups. The Group 0 did not send much data (Unique Bytes) into upstream direction whereas Group 1 contained those that had some significant data transfers in the upstream direction. More precisely, Group 1 was defined as the set of mobiles for which the total amount of upstream Unique Bytes was larger than 1 kB times the total number of TCP connections from that mobile. Figure 6 shows the maximum RTT of all TCP connections of a mobile against the maximum number of simultaneous connections observed from the same mobile for Group 0 (left figure) and 1 (right figure). The scales of axes are chosen to be the same for both plots in order to show that extremely large RTT values occur almost solely for Group 1 (significant upstream traffic), whereas very large number of simultaneous TCP connections occur for Group 0 (only downloading traffic). Because very large RTT values occur in Group 1, the context switching between transmitting and receiving packets sometimes causes problems.

It can be expected that observed maximum RTTs increase as the amount of simultaneous TCP connections increases. This is verified in Figure 7 (left), which shows an

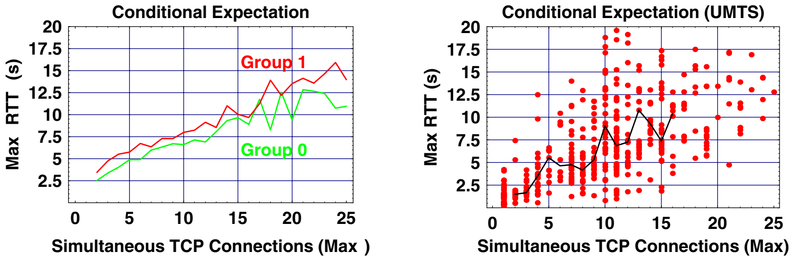


Fig. 7. Robust estimates of conditional expectations of maximum RTTs for GPRS (left) and UMTS (right). Right plot also shows individual samples of maximum RTTs.

estimate of the expectation of maximum RTT over all simultaneous flows, conditioned on the maximum number of simultaneous connections.

Our UMTS sample is not large enough to make the same division into groups as with GPRS. Figure 7 (right) shows the individual values of maximum RTTs as a function of the number of simultaneous TCP flows during a session (red dots), and the conditional expectation of maximum RTT over all simultaneous flows, conditioned on the maximum number of simultaneous connections (solid line). Self-congestion is also a problem for UMTS mobiles, although less serious.

5 Conclusions

Results on the variability of Round Trip Times (RTTs) of TCP flows in a Finnish operator's GPRS/UMTS network were given. Microscopic analysis of the RTTs of individual flows was performed on some selected flows, and we were able to detect, e.g., periodic behavior and rate changes in the radio channel. At the macroscopic level it was shown how bandwidth sharing at the mobile device itself seriously affects the RTTs of flows, both in GPRS and in UMTS.

References

1. Vacirca, F., Ziegler, T., Hasenleithner, E.: Large Scale Estimation of TCP Spurious Timeout Events in Operational GPRS Networks. In: COST 279. (2005)
2. Vacirca, F., Ricciato, F., Pilz, R.: Large-Scale RTT Measurements from an Operational UMTS/GPRS Network. In: First International Conference on Wireless Internet (IEEE WICON 05). (2005)
3. Benko, P., Malicsko, G., Veres, A.: A Large-scale, Passive Analysis of End-to-End TCP performance over GPRS. In: INFOCOM 2004. (2004)
4. Partridge, C., Cousins, D., Jackson, A., Krishnan, R., Saxena, T., Strayer, W.: Using Signal Processing to Analyze Wireless Data Traffic. In: ACM Workshop on Wireless Security, Atlanta, Georgia, USA, ACM (2002)
5. Kilpi, J., Lassila, P.: Statistical analysis of RTT variability in GPRS and UMTS networks. Technical report, VTT and TKK, `http://www.netlab.hut.fi/tutkimus/pannet/publ/rtt-report.pdf` (2005)

Control Plane Protection Using Link Management Protocol (LMP) in the ASON/GMPLS CARISMA Network[*]

Jordi Perelló, Eduard Escalona, Salvatore Spadaro, Fernando Agraz,
Jaume Comellas, and Gabriel Junyent

Optical Communications Group, Signal Theory and Communications Dept.,
Universitat Politècnica de Catalunya,
C. Nord D4-S107, Jordi Girona, 1-3, E-08034, Barcelona, Spain
{jperello, escalona, spadaro, agraz,
comellas, junyent}@tsc.upc.edu

Abstract. In the ITU-T ASON architecture, the control plane is responsible for providing intelligence to the network. The GMPLS paradigm pleads for a separation between the control plane and the forwarding plane. If the control plane is deployed disjoint from the forwarding plane, recovery mechanisms to ensure its proper operation are required. In this paper, on one hand, a quasi-associated mode backup control channel proposal is compared with a traditional associated 1:1 protection. On the other hand, extensions to LMP defined by the IETF are presented and evaluated to address both control channel and nodal failure recovery. The merits of the proposals are assessed by experimental results.

Keywords: ASON, GMPLS, LMP, protection.

1 Introduction

The increasing utilization of the Internet, the emerging applications that require large bandwidth, in conjunction with the nowadays high speed access networks, have put the current transport infrastructure in a tight spot. While designed to support circuit-based traffic, it shows its inefficiencies, mainly due to its static bandwidth provisioning when carrying IP and Ethernet based data traffic. It is proved that Automatic Switched Optical Network (ASON) architecture [1], defined by the International Telecommunications Union (ITU-T), has become a hopeful possibility to support the current data traffic explosion, which requires a high degree of flexibility. Among multiple functionalities, the ASON architecture accomplishes the requirement of fast, dynamic and flexible end-to-end bandwidth provisioning. This architecture relies on three well separated planes: an all-optical transport plane over which the light paths/connections are established, a control plane responsible for

[*] The work reported in this paper was supported in part by the Spanish Science Ministry through the Project "Red Inteligente GMPLS/ASON con Integración de Nodos Reconfigurables (RINGING)", (TEC2005-08051-C03-02).

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1182 – 1190, 2006.

creating, maintaining and deleting the requested connections, and a management plane with a whole network view, capable for requesting, supervising and tearing-down light paths as well as for managing both the transport and control plane. To meet the above mentioned requirements, the key entity in the ASON architecture is the control plane, which provides the necessary intelligence to the network. It supports the required routing and signaling information for dynamically creating the requested connections. The Generalized Multi-Protocol Label Switching (GMPLS) protocol set [2], defined by the Internet Engineering Task Force (IETF) in concordance with the ITU-T, arose as the preferred technology to implement control plane functions. GMPLS is an extension of the set of protocols designed for the MPLS technology and encompasses time-division (e.g. SONET/SDH, PDH, G.709), wavelength as well as spatial switching. The main GMPLS protocols are: Resource Reservation Protocol with Traffic Engineering Extensions (RSVP-TE) [3], Open Shortest Path First with Traffic Engineering Extensions (OSPF-TE) [4] and Link Management Protocol (LMP) [5][6]. Connection signaling tasks are performed by RSVP-TE. The OSPF-TE protocol is used to flood the state of all the node outgoing data links to all the other network nodes. In order to establish a connection, foremost a route to reach the destination node is calculated using the link state information provided by OSPF-TE. Afterwards, the request is forwarded by RSVP-TE to all the nodes involved in that connection. Upon reception of RSVP-TE Path/Resv messages, the required resources are reserved.

The whole of the control channels constitute the Data Communications Network (DCN) [1], whereby routing and signaling information is transmitted. LMP is defined by the IETF (hereafter standard LMP) as a new protocol with multiple functionalities, such as the management of the control channels between neighbors. Other functionalities of LMP are the correlation of the logical resources mapped over the physical existing resources between neighbors, link discovery, and fault isolation procedures.

This paper presents an enhanced control channel protection scheme in order to optimize the required control network resources. Moreover, some extensions to standard LMP are presented to properly perform control channel and nodal failure protection. Our proposals have been implemented and evaluated in the ASON/GMPLS CARISMA network, which relies on an out-of-fiber control plane.

The remainder of the paper is organized as follows: firstly the CARISMA network is described in Section 2 and the LMP protocol is overviewed in Section 3. Section 4 presents the proposed protection scheme and the results of its performance compared to the standard LMP. In Section 5, some extensions to the standard LMP are discussed in order to successfully perform control channel and nodal failure protection while minimizing the required control network resources. Finally, Section 6 concludes the paper.

2 The ASON/GMPLS CARISMA Network

The CARISMA project [7] was initiated in 2002 as an initiative to build a high performance Wavelength Division Multiplexing (WDM) based network to be used as a field-trial for the integration and evaluation of the current emerging innovative technologies. It is intended to provision bandwidth on demand while ensuring Quality of Service (QoS) between IP networks.

The CARISMA network (Fig. 1) implements the ASON architecture. Its transport plane is formed by three OADM capable optical nodes. These nodes are connected through two unidirectional fibers (working and protection fibers respectively) forming a dual ring topology. Each OADM is able to insert up to four WDM channels by means of transceivers (two of them tunable and two fixed) and to extract four channels of the twelve available in the ring. Each WDM channel is transparent to 2.5 Gbit/s and at least three of them to 10Gbit/s. The distance between nodes is about 35Km far, so the total ring length is more than 100Km.

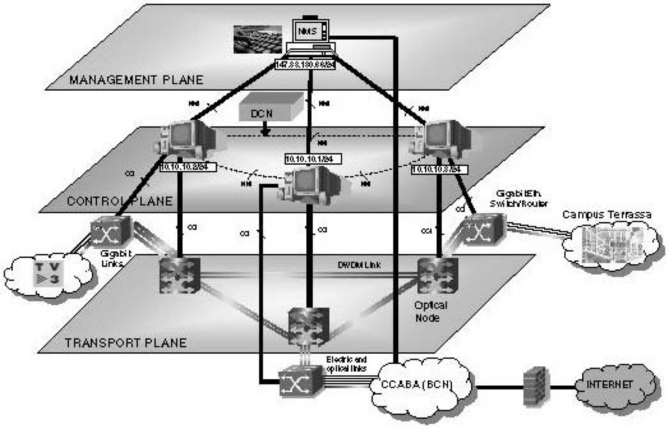


Fig. 1. The CARISMA network Architecture

The CARISMA network control plane has been deployed as an IP out-of-fiber network. Specifically, three Optical Connection Controllers (OCCs) are implemented using Linux-based routers, which run the GMPLS protocols. The three deployed OCCs are interconnected through Ethernet point-to-point links. The GMPLS paradigm intends to clearly separate the control plane from the data plane. Control channels can be in-fiber in-band, in-fiber out-of-band or out-of-fiber out-of-band. In the latter case, as in the CARISMA network, control plane liveliness is not associated to the data plane one, so control plane communication can be maintained alive upon a data plane failure and vice versa. Therefore, fault detection and protection mechanisms in addition to those existing in the data plane must be implemented also in the control plane. Since there is no association between the control channels and the data channels (e.g., as in MPLS), control channel protection can not be resolved using data plane protection mechanisms. In the CARISMA network we have implemented the LMP protocol to maintain both control channel connectivity and link properties between two data plane adjacent neighbors.

Finally, the CARISMA network management plane is formed by the Network Management System (NMS), implemented as a web application facilitating network administration through Internet.

3 Link Management Protocol Overview

In order to enable the communication between nodes for signaling, routing and link management purposes, control channels must be established between any pair of nodes. The LMP protocol has been defined to fulfill control channel management and also to perform additional functionalities. The four functionalities proposed to be done by LMP are: control channel management, link property correlation, link connectivity verification and fault management. The first two are mandatory when implementing LMP, whereas the rest are optional.

Control channel management is related to two procedures, namely the control channel establishment and the maintenance between LMP neighbors. Specifically, a hello-based keep-alive mechanism is used to maintain control channel connectivity. These procedures begin with a negotiation phase, where the control channel is established and the keep-alive mechanism intervals negotiated. The use of a keep-alive mechanism takes crucial importance when lower-layer mechanisms are not able to detect control channel failures (e.g., out-of-fiber control plane). Hello messages are transmitted every *HelloInterval*. If no hellos are received in a *HelloDeadInterval*, control channel connectivity is declared lost.

Link property correlation deals with the synchronization of the properties of the defined TE links between adjacent neighbors, where a TE link is a logical aggregation of various data links defined between neighbors. In fact, those properties include both the TE link local and remote identifiers and the characteristics of all the data links contained in that TE link. This process is achieved by sending *LinkSummary* messages to a neighbor, which contain all the properties referent to a TE link towards that neighbor. The information contained in a *LinkSummary* message can be agreed or disagreed by responding with a *LinkSummaryAck* or a *LinkSummaryNack*. If the information comprised in a *LinkSummary* message is set to non negotiable, it is forced to be accepted. Link property correlation procedures must be done before a TE link is considered ready to transmit traffic. Furthermore it can be periodically performed.

Link connectivity verification is required to test the physical connectivity of the data links, and also to dynamically learn the TE link and data link ID associations, so it can be used for link discovery purposes. On the other hand, fault management is intended to be used to isolate data link and TE link failures. It becomes greatly useful if physical circuits are established upon an all-optical transport plane. In such environment, Loss of Light (LoL) fault detection mechanisms do not apply properly, since LoL alarms are detected by all the downstream nodes from the point where the failure has occurred. These two mechanisms are out of the scope of this paper.

4 Protection Scheme for Control Channel Failure

The standard LMP does not focus on any control channel protection schemes. In fact, in [5], control channel protection schemes such as 1+N (i.e., sending signaling and routing information through one or more control channels towards the same neighbor at the same time) or dedicated 1:1 (i.e. having a standby control channel waiting to

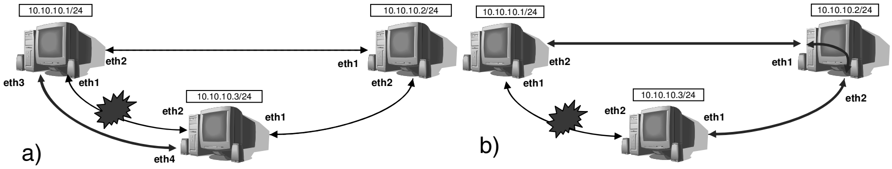


Fig. 2. Control channels protection schemes: a) Dedicated 1:1 backup control channel, b) Proposed control channel scheme: the backup channel is established through the alternative disjoint to the failure route

become active upon working control channel failure (Fig. 2a)) are just mentioned. In this Section, we focus on how it is possible to take advantage of the LMP control channel management procedure for the out-of-fiber control plane protection. Generally speaking, according to [8], three kinds of control channels can be established: associated, quasi-associated and non-associated. Associated control channels directly interconnect two physically adjacent neighbors, whereas quasi-associated and non-associated control channels indirectly interconnect two physically adjacent nodes through a pre-determined route or following an undetermined route respectively.

The out-of-fiber CARISMA network control plane is based on a bidirectional ring topology. Therefore an alternative route which avoids a determined failed control channel is always available. In this context, establishing a disjoint quasi-associated backup control channel that surrounds the affected control channel is really advantageous. This scheme (Fig. 2b) is better than the associated mode possibilities (e.g., 1+1 protection, 1:1 dedicated protection) in terms of both resource and computational cost savings. In fact, on one hand, if more than one active control channel between neighbors is used, redundant control traffic packets have to be sent over the redundant control channels. Moreover, the redundant packets have to be discarded upon reception. Although this solution presents a nearly zero switching time, its required packet overhead, which increments nodal computational cost, makes it inappropriate to be applied to protect the DCN, where traffic recovery times are not as restrictive as in the transport plane.

On the other hand, dedicated 1:1 protection scheme does not require the computational cost as the previous option. However, it increments the number of resources required to implement control plane protection in contrast with the quasi-associated scheme which is proposed in this paper. Fig. 2 presents the two evaluated protection schemes.

This paper compares the here proposed quasi-associated protection scheme and the dedicated 1:1 one. The comparison has been done in terms of protection switching time, which is the time required to set up the backup control channel after the working control channel failure detection. It includes the backup control channel negotiation phase and the working control channel interface shutdown. Such protection switching time results have been experimentally obtained by using different LMP hello

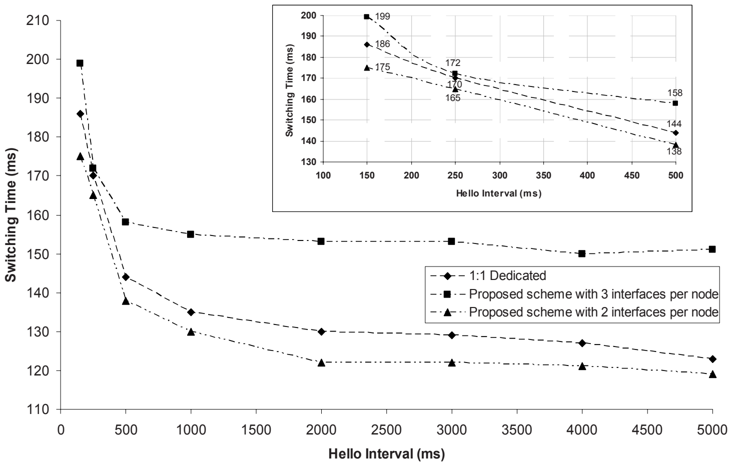


Fig. 3. Obtained control traffic switching times

intervals. Choosing the correct hello intervals depends on the control plane protocols which are running. Upon a control channel failure, too high hello intervals can result in blocked connection requests, outdated node Traffic Engineering Databases (TED) or even RSVP-TE state loss. *OSPF Hello* messages and RSVP-TE retransmission times are in the order of tens of seconds, so a 5 second hello interval seems to be sufficient. Nevertheless, in order to minimize blocked connection requests and Link State Advertisement (LSA) losses, lower hello requests are required. Fig. 3 illustrates the obtained control traffic switching times function of the used HelloInterval value. Each point has been obtained as the mean of a statistical relevant number of results.

It can be seen that control channel traffic switching times increase as hello intervals decrease. The same happens for node overload. The proposed protection scheme has lower control channel switching times than the dedicated 1:1 one. This is due to the fact that the nodes which implement dedicated 1:1 protection have to maintain an additional interface, causing an increased computational cost. To verify it, control channel switching times obtained with the proposed scheme but in a scenario where each node maintain not two but three interfaces have been also included in Fig. 3. The LMP standardization proposes 150 ms for *HelloInterval* and 500 ms for *HelloDeadInterval*. However, values of 500 ms and 1500 ms for them are used in the CARISMA network. While avoiding signaling and routing operation to be disrupted, they involve lower CPU costs, reduce the traffic over the DCN and they are totally applicable in a metropolitan environment with low incoming call volumes.

Using the proposed intervals, we obtain significantly lower control plane restoration times compared with the tens of seconds of IP dynamic routing, and even with in-fiber management solutions, such as the one proposed in [9]. This is due to the fact that, since the control plane is decoupled from the transport plane, no transport plane configurations should be modified upon a control plane failure, which spares TED updates.

5 LMP Extensions for Control Channel and Neighbor Node Controller Failures

In this Section, some extensions to the standard LMP to be used for both control channel and neighbor node failure situations are presented and evaluated. The standard LMP does not provide specific mechanisms for control plane network protection. On one hand, as above discussed, redundant control channel connectivity has to be avoided. On the other hand, criterions to be applied prior to LMP graceful restart procedures to properly distinguish nodal from control channel failures, take crucial importance.

According to the standard LMP, when no hello messages are received in a *HelloDeadInterval*, the node assumes that the working control channel is down. The node then set both the working and the backup control channel, which was in a standby state, to the negotiation state (i.e., the node sends *Config* messages towards its neighbors). Maintaining the failed working control channel in the negotiation state has its pros and cons. In fact, allowing the automatic control channel re-establishment, once the failure has been repaired, entails, for one hand unnecessary node controller computational costs and, on the other hand, redundant control channel connectivity.

To overcome both the computational cost and redundant connectivity problems, we propose a mechanism, which is an extension of the standard LMP, in order to tear down the backup control channel upon working control channel re-establishment. It uses the *ControlChannelDown* flag functionalities [5], which actually allow to gracefully take down a certain control channel. Fig. 4 shows the state diagram of our proposal. We consider three control channel states: Up, Going Down and Down [5]. The Up state is the operational state, wherein the hello-based keep-alive mechanism is performed. On the contrary, the Down state is the initial state, where the control channel is on standby and no attempts to bring up the control channel are made. A control channel passes from the Up to the Going Down state when an administratively control channel tear down is desired. In this state, the node sets the *ControlChannelDown* flag to 1 in all messages it sends.

When the working control channel connectivity is re-established after the failure is repaired, the event which indicates the first received valid hello message (*evHelloRcvd*), meaning a successful negotiation phase, is caught. Upon this event, the backup control channel passes from the Up state to the Going Down state, since its functionality is no more needed due to the fact that the working control channel is again fully operating. If a message is received with the *ControlChannelDown* flag set to 1 or no messages with this flag set to 1 are received in a *HelloDeadInterval*, the backup control channel is considered down. Fig. 4 shows that proposed functionality.

The remainder of this section deals with LMP session recovery after a node failure. It can be possible that the failed node takes long to restart, making useless the negotiation state of the backup control channel, since the working control channel is healthy and has to be re-established once the failure is overcome. When a neighbor node controller failure occurs, LMP graceful restart mechanisms [5] should be applied to re-synchronize the properties of the defined TE links between adjacent neighbors.

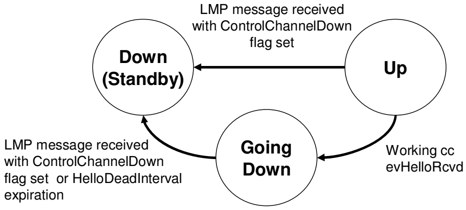

LMP message received with ControlChannelDown flag set
Down (Standby)
Up
Going Down
LMP message received with ControlChannelDown flag set or HelloDeadInterval expiration
Working cc evHelloRcvd

Fig. 4. Proposed backup control channel performance state diagram to avoid redundant control channel connectivity

In order to avoid this excess of computation costs related to maintaining the backup control channel into a negotiation state upon a neighbor node controller failure, we propose a method so the neighbor nodes detect if the failure is a link failure or a controller failure prior to LMP graceful restart procedures. When the node initiates the backup control channel negotiations, a timer set to three times *ConfigRetryInterval* is initiated. If no *Config*, *ConfigAck* or *ConfigNack* messages are received before this timer expiration, a neighbor controller failure is considered. Then, the backup control channel is set again to standby and the *Config* messages will just be sent through the working control channel. This is done until the reception of a *Config* message from the failed neighbor with the Restart flag set to 1. In that way, this proposed mechanism allows nodal failure distinction from control channel failure before the restart and reduces nodal computational costs by removing the backup control channel negotiation state. The proposed performance is shown in Fig. 5. The *Config* retry interval and *ConfigDeadInterval* used in the CARISMA network have been 500 ms and 1500 ms respectively.

To complete the LMP session restoration upon a neighbor controller failure, a *LinkSummary* message with no negotiable information is sent towards the restarted node for every TE link established between them. Graceful restart procedures avoid

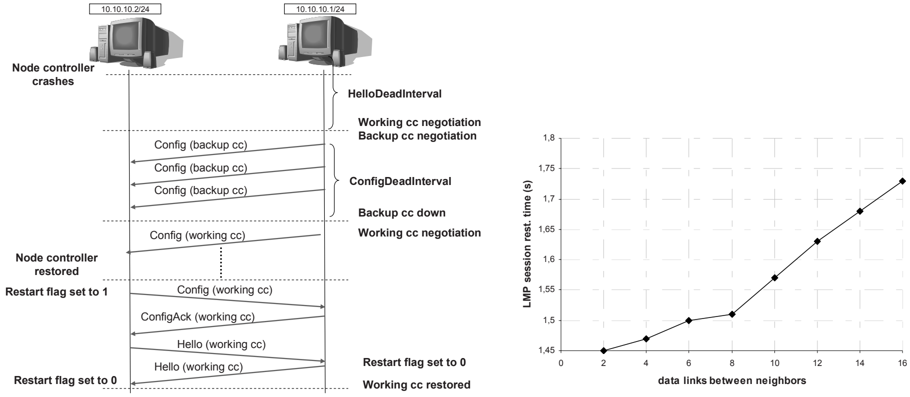


Fig. 5. Proposed method for LMP session restoration and LMP session restoration times function of the established data links between neighbors

TE link and data link parameter misconfigurations once the node is restarted. LMP session restoration times function of the defined data links (i.e lambdas) between neighbors are also depicted in Fig. 5, where the obtained points are the mean of a statistical relevant number of results. They are measured since the neighbor node controller is restored, to the ending of LMP graceful restart procedures, obtaining values close to 1,5 s.

6 Conclusions

This paper presents an alternative backup control channel protection scheme in order to minimize the required resources to perform DCN protection in the out-of-fiber control plane of the ASON-based CARISMA network. Its functionality in terms of control traffic switching time has been evaluated and compared with the 1:1 dedicated protection. The obtained results not only show the feasibility of our proposal, but also reflect improved control traffic switching times in a metropolitan environment (with a low number of nodes and links) compared with the dedicated option.

Some extensions to the standard LMP have been proposed in order to properly perform both control channel failure recovery and nodal LMP session recovery optimization. Such extensions allow, on one hand, to avoid redundant control channel connectivity and, on the other hand, to differentiate between control channel and node controller failures. The latter implies the reduction of the computational costs for the node controllers. Upon nodal failure recovery, LMP graceful restart procedures have to be performed to re-synchronize the state of the TE links defined between neighbors.

The experimental results demonstrates that LMP with some extensions can be a useful way for providing control plane protection in ASON based out-of-fiber control plane environments.

References

1. ITU-T Recommendation G.8080: Architecture for the Automatically Switched Optical Network (ASON), 2001.
2. Mannie, E. (ed.): Generalized Multi-Protocol Label Switching Architecture. IETF RFC 3945, 2004.
3. Berger, L. (ed.): Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions. IETF RFC 3473, 2003.
4. Katz,D., Kompella, K., Yeung,D.: Traffic Engineering (TE) Extensions to OSPF Version 2. IETF RFC 3630, 2003.
5. Lang, J. (ed.): Link Management Protocol (LMP). IETF RFC 4204, 2005.
6. Fredette, A.(ed.), Lang, J. (ed.): Link Management Protocol (LMP) for Dense Wavelength Division Multiplexing (DWDM) Optical Line Systems. IETF RFC 4209, 2005.
7. CARISMA (Conexión y acceso a RedIRIS2 mediante anillo óptico multicanal) Project, http://carisma.ccaba.upc.edu
8. Young, K.: Requirements for the Resilience of Control Plane. IETF Internet draft draft-kim-ccamp-cpr-reqts-01.txt, 2005.
9. Muñoz, R., et al. : Experimental GMPLS fault management for OULSR transport networks, OSA/IEEE Optical Fiber Communications/SPIE National Fiber Optic Engineers Conference (OFC/NFOEC 2005).

A Novel Resource Allocation Scheme for Reducing MAP Overhead and Maximizing Throughput in MIMO-OFDM Systems

Chung Ha Koh, Kyung Ho Sohn, Ji Wan Song, and Young Yong Kim

Dept. of Electrical and Electronic Engineering,
Yonsei University,
Seoul, Korea 120-749
{ski244, heroson7, wanbabo, y2k}@yonsei.ac.kr

Abstract. We propose a novel resource allocation scheme, which can reduce MAP overhead and maximize the throughput in the MIMO-OFDM systems. In the message based broadband access system, we need to minimize the MAP overhead since the excessive MAP overhead causes degradation of system throughput. Increasing the size of resource allocation unit can reduce MAP overhead. However, multiuser diversity gain becomes smaller as the size of re-source allocation unit increases. Therefore, we investigate joint optimization between multiuser diversity gain and MAP overhead size. Using the proposed scheme, we can reduce MAP overhead size as well as achieve high throughput.

1 Introduction

Message based multiplexing is becoming more prominent than channel based multiplexing in the next generation communication systems like 802.16 Broadband Wireless Network [1]. The system operating with channel based multiplexing gives user monopolistic rights to use the channel resource when a session is opened. Channel based multiplexing employs predivided area as the resource allocation unit. In contrast, the system exploiting message based multiplexing allocates the resource using flexible resource allocation unit. Message based multiplexing is well matched with the burst nature of data traffic, and it makes the system use radio resource effectively.

Message based multiplexing system has to inform all users of the results from resource allocation by using a frame message because the results of resource allocation are changed every frame. However, message based multiplexing has very large message overhead since every user is given all information pertaining to the allocation results. In the Broadband Wireless Network system, the base station (BS) broadcasts a MAP message which is appended to the front part of each frame in order to transfer the allocation information. Furthermore, the users are needed to receive the MAP message reliably, so the MAP message must be transmitted with low order modulation and heavy coding. Therefore, the transmission time of the MAP message becomes longer and system throughput reduces relatively with the decrease of data transmit time.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1191–1196, 2006.

Our main focus is on the allocation unit of frequency resource. We assume that a subband consists of several subchannels and regard it as the frequency resource allocation unit. Note that the subband size means the number of subchannels in it. When the subband size becomes larger, the MAP overhead size becomes smaller since the number of allocation is decreased. On the other hand, as the allocation unit increases, the multiuser diversity gain decreases because the capacity of a subband is determined by minimum channel capacity of subchannels in it. Therefore the potential to exploit higher data throughput introduces a trade-off problem between the size of the MAP message and multiuser diversity gain.

After analyzing the relation between multiuser diversity gain and MAP overhead size, we propose a new scheme of determining an optimum resource allocation unit. In this paper, we have found the optimum subband size to maximize the system through-put. Our simulation results shows that the proposed algorithm tends to outperform the throughput in terms of maximizing throughput.

The rest of this paper is begins with the performance analysis of trade-off between multiuser diversity gain and MAP overhead. Section 3 shows simulation results, and conclusions are drawn with some final remarks in Section 4.

2 Problem Formulation and Analysis

In this section, we find the optimum subband size which maximizes the system throughput considering trade-off relationship between multiuser diversity gain and MAP overhead size. Firstly, to simplify our analysis we assume that the all subchannels have an independent and identical distribution (i.i.d.) channel capacity. Then, we expand the results to the situation in which each user has different channel capacity and propose the heuristic algorithm for that case.

2.1 I.I.D. Channel Capacity Case

To simply our analysis, we consider i.i.d. subchannel capacity distribution and best user selection scheduling scheme. Let us assume $X_{k,n}$ is an i.i.d. sequence of random variable for the capacity given subchannel n, user k. If we make a subband by binding p subchannels and use it as a unit of resource allocation, Y_k, a subband capacity of user k, is

$$Y_k = mean\{X_{k,1}, X_{k,2}, \cdots, X_{k,p}\} \tag{1}$$

X has a Gaussian distribution under the condition of sufficient number of antennas in the MIMO system [2],[5], so the average value of an X sequences and Y have also a Gaussian distribution. If the mean value of X is μ_X and the variance value is σ_X^2, then the mean value of Y, denoted as μ_Y, is μ_X, and the value of variance Y, denoted as σ_Y^2, is $(\sigma_X/p)^2$. When we consider the best user selection scheduling that allocates the subband to maximum subband capacity user, the average throughput is calculated as follows [5].

$$M = \max\{Y_1, Y_2, \cdots, Y_K\} \tag{2}$$

$$E[M] = \mu_Y \ + \sqrt{2\sigma_Y^2 \log K} \ = \ \mu_X \ + \frac{\sqrt{2\sigma_X^2 \log K}}{p} \tag{3}$$

Here, $\sqrt{2\sigma_X^2 \log K}/p$ is the multiuser diversity gain from scheduling leading to high increase of throughput. When the value of p gets larger, the amount of multiuser diversity gain gets smaller. This is because selecting max capacity in each subchannel can obtain larger scheduling gain than selecting max capacity among the mean capacity of several binding subchannels.

The transmission time for the MAP message is formulated as follows. Downlink MAP message consist of various elements [1]. The DL-MAP is specified by the MAP IE, which represents the allocating result that exploit the same modulation order and coding rate. MAP message transmission time depends on the number of MAP information elements (IEs) which indicate the number of allocated users in a frame.

When we allocate resources in subbands which is the binding of several subchannels, the number of IEs decreases as shown in Fig. 1. Under our assumption that all the subchannels are independent and identical, the probability for selecting user k in one subband, Pr_k, is all the same for all users, $1/K$, where K is the number of total users. Therefore, the number of IEs is derived as follows, where N is the total number of subchannels and p is the subband size.

$$E[No.\ of\ IE] = \sum_{k=1}^{K} \frac{N}{p} \cdot Pr_k = \frac{N}{p} \tag{4}$$

Let r be the constant value for transforming units from bit to sec, which influences on the modulation order and coding rate. If we ignore the fixed number of bits of DL-MAP, we can obtain T_{MAP}, the MAP transmission time by multiplying the number of IE by constant r. Since one frame length, T_{frame}, can be divided into overhead transmission time and data transmission time, the data transmission time, T_{data}, is calculated by subtracting T_{MAP} from T_{frame}. Therefore, we can derive the total throughput as the product form of $E[M]$ and T_{data}. We can get the optimal subband size p^* to maximize the system throughput.

$$p^* = \arg\max_p \left(\mu_x + \frac{\sqrt{2\sigma_x^2 \log K}}{p} \right) \cdot \left(T_{frmae} - \frac{N}{p} \cdot r \right) \tag{5}$$

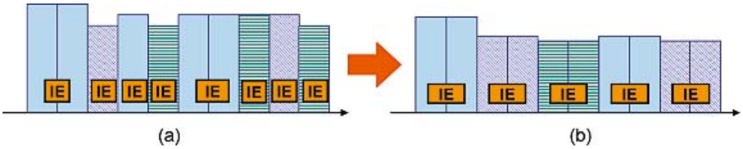


Fig. 1. Comparision of the number of IEs using (a) subband size=1 and (b) subband size=2

2.2 Different Capacity of Each User Case

In the previous subsection, we showed that optimum subband size in the case of i.i.d. subchannel. We now present numerical results in the more practical case where each user's subchannel has a different capacity distribution from other user's subchannel capacity. In this situation, it is advantageous to determine subband sizes that are different from each other. Let us assume p_k is a subband size of user k, and Y_k is the subband capacity of user k, $Y_k = mean\{X_{k,1}, X_{k,2}, \cdots, X_{k,p_k}\}$. Then, we can obtain the multiuser diversity gain as follows [6].

$$\begin{aligned}
E[M] &= \sum_{k=1}^{K} E[Y_k | Y_k \text{ is maximal}] \\
&= \sum_{k=1}^{K} \left\{ \int_{-\infty}^{\infty} y \frac{\Pr(Y_k \text{ is maximal} | Y_k = y)}{\Pr(Y_k \text{ is maximal})} \frac{1}{\sqrt{2\pi}\sigma_k/p_k} e^{-\frac{(y-\mu_k)^2}{2(\sigma_k/p_k)^2}} dy \right\} \\
&= \sum_{k=1}^{K} \left[\frac{1}{\Pr_k} \int_{-\infty}^{\infty} \frac{y}{\sqrt{2\pi}\sigma_k/p_k} e^{-\frac{(y-\mu_k)^2}{2(\sigma_k/p_k)^2}} \prod_{\substack{i=1 \\ i\neq k}}^{K} \left\{ 0.5 + 0.5erf\left(\frac{y-\mu_i}{\sqrt{2}\sigma_i/p_i}\right) \right\} dy \right]
\end{aligned} \tag{6}$$

Alike notation of previous section, M is the random variable of the selected user who has the largest subband capacity, and μ_k, σ_k^2 represents the mean and variance value of user k' subchannel capacity respectively. Then, data transmission time is like below.

$$T_{data} = T_{frame} - \left(\sum_{i=1}^{K} \frac{N}{p_i} \cdot \Pr_i \right) \cdot r \tag{7}$$

Given (6) and (7), the total throughput of this system is given by product of $E[M]$ and T_{data}, where Pr_k is defined as the probability which user k is selected. Pr_k can be written as

$$\begin{aligned}
\Pr_k(\mu, \sigma) &= \Pr(Y_k \; is \text{ maximal}) \\
&= \int_{-\infty}^{\infty} \frac{1}{\sigma_k/p_k\sqrt{2\pi}} e^{-\frac{(x-\mu_k)^2}{2(\sigma_k/p_k)^2}} \prod_{\substack{i=1 \\ i\neq k}}^{K} \left(0.5 + 0.5erf\left(\frac{x-\mu_i}{\sqrt{2}\sigma_i/p_i}\right) \right) dx
\end{aligned} \tag{8}$$

To find the optimum solution which maximizes the total throughput, $\overrightarrow{p}^* = [p_1^*, p_2^*, ..., p_K^*]$, is difficult because it is a non-linear optimization problem. Therefore, obtaining an optimal solution is very difficult. That is why we proposed heuristic algorithm MAP Reduced Resource Allocation (MRRA)

2.3 MAP Reduced Resource Allocation

We assume the Gaussian MIMO channel capacity with mean vector $\overrightarrow{\mu} = [\mu_1, \mu_2, ..., \mu_K]$ and standard deviation vector $\overrightarrow{\sigma} = [\sigma_1, \sigma_2, ..., \sigma_K]$. In the MRRA algorithm, pre-calculated values of the optimum subband size according to the capacity distribution are used. We consider the i.i.d. channel assumption in order to simplify the problem. Namely, we assume that other users' channel capacity are the same as user k, and then calculated the optimum subband size of user k, p_k.

$$p_k = \arg\max_{p} \left(\mu_k + \frac{\sqrt{2\sigma_k^2 \log K}}{p} \right) \cdot \left(T_{frame} - \frac{N}{p} \cdot r \right) \tag{9}$$

We can determine the optimum subband size using (9). It is effective because the multi-dimensional search is not necessary. In addition, the adaptive scheme according to various capacity distributions can be exploited. However, if the subband size of each user is different from each other, a problem may occur in the resource allocation steps since the general resource allocation schemes are based on the same allocation unit size. Therefore, we consider integrating each user's subband size to one value by averaging. We have also proved the performance of this heuristic algorithm by simulation.

3 Simulation Results

We now describe some simulation experiments which were conducted to quantify the performance gains from proposed subband size determining scheme. Consider a 4x4 MIMO-OFDM system operating with 40MHz bandwidth and 256 subchannels under Rayleigh flat fading channel. Each time frame is 1ms in length, and single-cell MIMO-OFDM system model is used. Let α be the initial MAP loading parameter that the proportional value of MAP message transmission time over a frame length. Accordingly, the initial MAP loading α is a real value between zero to one.

In order to evaluate the performance of the proposed MRRA scheme, it is compared with other resource allocation schemes. Here, we consider the comparison

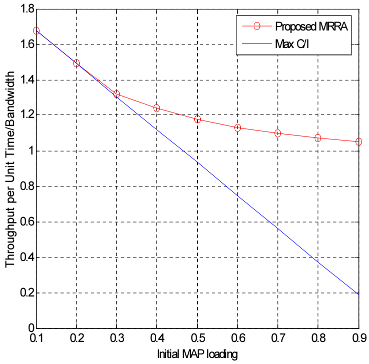


Fig. 2. Normalized throughput vs. subband size (i.i.d. case)

with Max C/I resource allocation method. In the Max C/I resource allocation scheme, the system allocates a subband to the user who has the best channel capacity of that subband. The two schemes are compared via computer simulation using the same single-cell OFDM system model. Users' positions are all random, so that the capacity distribution of all users is different from each other.

Fig. 2 presents throughputs under the proposed MRRA scheme and max C/I scheme, respectively as the initial MAP loading increases. The proposed scheme achieves higher throughput when the MAP overhead get larger. Considering throughput, the MRRA scheme performs as high as 17% in the case of $\alpha =$ 0.5. In addition, when α is 0.8, the proposed MRRA scheme exhibits twice the throughput of Max C/I scheme. This throughput gain results from joint optimization of multiuser diversity and MAP transmission time.

In general, we observed that the performance gains from the MRRA scheme depend on the initial MAP loading. When there is heavy initial MAP loading, proposed scheme may help considerably improve performance.

4 Conclusions

In this paper, we have considered maximizing downlink throughput for MIMO-OFDM systems by exploiting joint optimization between multiuser diversity and MAP overhead. In particular, we focused on determining the optimum subband size to maximize the system throughput. Simulation results show that the proposed MRRA scheme outperforms a Max C/I scheme in terms of throughput due to the optimality of our scheme. In the future MIMO-OFDM systems, the proposed MRRA algorithm should take into account specific resource allocation scheme in order to guarantee the QoS, which needs further investigation.

References

1. IEEE 802.16-2004, "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems," Oct.01, 2004
2. Peter J Smith and Mansoor Shafi, "On the Gaussian Approximation to the Capacity of Wireless MIMO System," Proceedings of IEEE ICC 2002, pp. 406-410, New York, May 2002
3. G.J. Focshini, "Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multi-Element Antennas," Bell Labs Technical Journal, pp.41-59, Oct.1996.
4. V.L. Girko, "A Refinement of the Central Limit Theorem for Random Determinants," The-ory of Probability and its Application, vol.42, no.1, pp.121-129, 1997.
5. B.W. Hochwald, T.L.Marzetta, and V. Tarokh, "Multi-Antenna Channel-Hardening and its Implications for Rate Feedback and Scheduling," IEEE transactions on Information Theory, vol. 50, no.9, pp.1893-1909, Sep. 2004
6. Athanasios Papoulis and S. Unnikrishna Pillai, "Probability, Random Variables and Stochastic Processes," 4th ed. Mc Graw Hill

Secure Routing Using Factual Correctness

Muthusrinivasan Muthuprasanna and Govindarasu Manimaran

Iowa State University
{muthu, gmani}@iastate.edu

Abstract. The routing protocols in use today operate on implicit trust among the different routers. Specifically, the distance vector routing (DVR) protocols compute routing tables in a distributed manner, based on this implicit trust. This trust model however fails to ensure the factual correctness of the routing updates, which is very critical for secure routing. We propose a neighbor update propagation model to ensure factual correctness and detect malicious activity by any subverted router. We also propose a secure DVR protocol based on this model using simple cryptographic primitives, and with minimal operational overhead.

1 Introduction

The research on routing protocols has proceeded along three different lines - Distance Vector (RIP), Path Vector (BGP) and Link State Protocols (OSPF). Malicious attacks, unintended misconfigurations, and simple/byzantine failures can lead to poisoning of the routing tables and result in drastic consequences [1]: AS7007 [2], AS3561 [3] incidents, etc. We focus on the distance vector routing (DVR) protocols here. They are simple efficient distributed algorithms that can be hijacked by modifying, replaying or deleting routing updates using subverted routers and links [4], and can result in sub-optimal routing, network partitioning, DoS attacks, etc. [5]. The desirable properties of secure routing protocols include: quick convergence, scalability, consistency, data integrity, origin authenticity and factual correctness [6]. Our primary focus here has been to ensure the factual correctness of the routing updates and to design a secure DVR protocol.

2 Related Work

Proposals to secure DVR protocols have been along two different lines - lightweight solutions providing limited security guarantees, and computationally-intensive solutions providing much higher security guarantees. The use of sequence numbers [4], consistency check (CC) [4] and PAIR [5] algorithms, digital signatures [7] and Intrusion Detection techniques [8] provide limited security guarantees. On the other hand, S-RIP [9] guarantees minimal factual correctness by using a reputation management framework, while SEAD [10], [11] uses a one-way hash chain to provide update security. In [12], a generic framework to implement secure protocols using a topological map has been proposed.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1197–1203, 2006.

Our contribution in this paper is two-fold. Firstly, we propose a neighbor update propagation (NUP) model that can ensure factual correctness of the routing updates in an adversarial environment. Secondly, we design a light-weight secure DVR protocol based on the above model. Here, we assume an attack model where the attacker is free to choose any attack technique and/or attack agent.

3 Factual Correctness Concept

The main idea of the proposed model is as follows: a routing update generated by a sender is as usual sent to its one-hop neighbors (receivers), and additionally also to its two-hop neighbors (verifiers); such that a subsequent update triggered and sent by the receiver to its one-hop neighbors (sender, verifiers as above) can easily be evaluated by them for consistency, as they have the precise knowledge of the trigger and the resulting update.

3.1 Neighbor Update Propagation (NUP) Model

The DVR protocol enables every router to determine in a distributed manner, the next hop router for every other destination in the network. This simple protocol takes in as inputs the current routing table and the received routing updates and outputs a new routing table. Thus it is independent of the location/identity of the router and its neighbors; and given the same set of inputs to any other router, it would spew out the same output. We exploit this feature in our NUP model. We now define a group as consisting of a node (sender), a single one-hop neighbor (receiver) and all the two-hop neighbors directly linked to the receiver (verifiers). Thus there exist as many groups as the number of routers in the network. In Fig. 1, node 2 is a receiver in Group 1 and sender in Group 2, while node 4 is a verifier in Group 1 and receiver in Group 2.

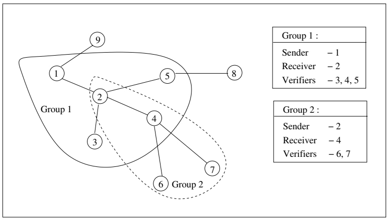


Fig. 1. Groups in NUP model

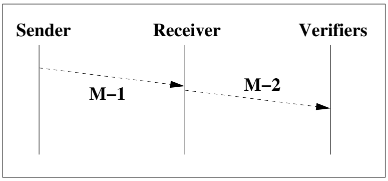


Fig. 2. Messaging in NUP model

The NUP model maintains a snapshot of the receiver's routing table at all its neighbors (verifiers). Any update that a sender (some neighbor) sends to the receiver is then forwarded to all the verifiers, who then maintain an updated cache of the receiver's routing table. The update that is subsequently generated by the receiver is validated by the verifiers for consistency with the cached routing table. Thus, the basic principle is to provide the verifiers the precise knowledge of what they are supposed to receive, so that they can detect any malicious update sent by the receiver at any time. This, when implemented for every group in the network, provides the necessary factual correctness security guarantees. Consider the sample topology in Fig. 1. The routing table of node 4 is cached by nodes 2, 6 and 7. Now when node 2 sends an update to node 4, it is also propagated to nodes 6 and 7. Nodes 2, 4, 6 and 7 now update (cache of) node 4's routing table appropriately. If node 4 sends any update in future, nodes 2, 6 and 7 verify its validity by comparing it with the cached routing table for consistency. This consistency check ensures the factual correctness of node 4's updates.

3.2 Factual Correctness Guarantee

It is the process by which a router ensures whether the update generated by its neighboring router is actually the same that any trusted or well-behaved router in that place would have generated. This, when performed by every router in the network, can ensure that no router can act maliciously by altering any update in a manner other than it is supposed to.

The two messaging rounds of the NUP model are as shown in Fig. 2. In Round 1, the sender sends a routing update to the receiver ($M1$). In Round 2, the receiver forwards that update to the verifiers (its other one-hop neighbors) ($M2$). Additionally, we identify two types of routing updates sent distinctly, *source updates* and *forwarded updates.* Consider the protocol operation in Group 2 in Fig. 1.

1. A forwarded update is triggered by a previously received routing update, and hence its factual correctness can be trivially verified as explained above.
2. If the source update indicates that a new link is now operational, router 4 (receiver) can employ cryptographic verification (HTC computation, as explained later) to detect that change. eg. new link from router 2 to router 9.
3. If the source update indicates a link weight change, the change will also occur in the update that the other neighbors (sender) send to router 2 (receiver) and router 4 (verifier) in Group 1, and hence can be validated. eg. weight change of link connecting routers 1 and 2.
4. If the source update indicates that a link has gone down, there is no way for router 4 (receiver) to verify it as router 2 (sender) could explicitly drop all packets coming from that neighbor. eg. link failure between routers 2 and 5. Hence the receiver has to trust the claim, but it is not a problem - if it were a malicious router, it would be limiting its own connectivity in the network and hence implicitly checking the spread of the malicious routing information. However, connectivity could be lost in the network and could unavoidably affect routing.

Any malicious activity in a group can be detected by the other routers in that group; and as every router is a sender, receiver, or verifier in every group it belongs to, any single-router hijack is easily identifiable and hence the NUP model is *single-router hijack resistant.* However, the problem of ensuring packet forwarding in accordance with these secured routing tables is beyond the scope of this paper.

4 NUP-Based Secure DVR Protocol

The immediate problem now faced is, how would a sender send a routing update to a verifier to which it is not directly connected, without the receiver maliciously altering it? To ensure data integrity and origin authenticity for these updates, we propose the Hash Table-Chain (HTC) construct, using simple cryptographic primitives - collision-resistant one-way hash functions and symmetric encryption.

4.1 Hash Table-Chain (HTC)

The basic principle here is to morph every message appropriately using a one time pad, generated by the HTC. As the HTC is mutually agreed upon by the sender and a certain verifier, any alteration of the update message by the receiver can be detected. We assume the existence of a Public Key Infrastructure (PKI) in the network. All group members agree upon a pairwise mutually shared key (a random nonce) using public-private keys, say a_1 as in Fig. 3. They then use cryptographic (one-way) collision-resistant non-invertible hash functions H and M to generate a hash chain (Eqn. 1) of length k and a hash table (Eqn. 2) of size n using an offset padding respectively. For the first routing update, the sender uses the hash table corresponding to a_k as a one-time pad to morph the routing update using symmetric encryption. For all subsequent updates, it uses hash tables corresponding to $a_{(k-1)}$, $a_{(k-2)}$, ... ,a_1 respectively. Once the hash chain has been exhausted, the nodes re-negotiate another shared key using public-private keys, and the process continues. The symbols $||$ and $\oplus$, represent the concatenation and symmetric encryption operations respectively.

$$a_1, H(a_1), H^2(a_1), ..., H^k(a_1) \ : \ H^i(a_1) = H(H^{(i-1)}(a_1)), \ H^0(a_1) = a_1 \quad (1)$$

$$M(1||a_i), M(2||a_i), M(3||a_i), ..., M(n||a_i) \quad (2)$$

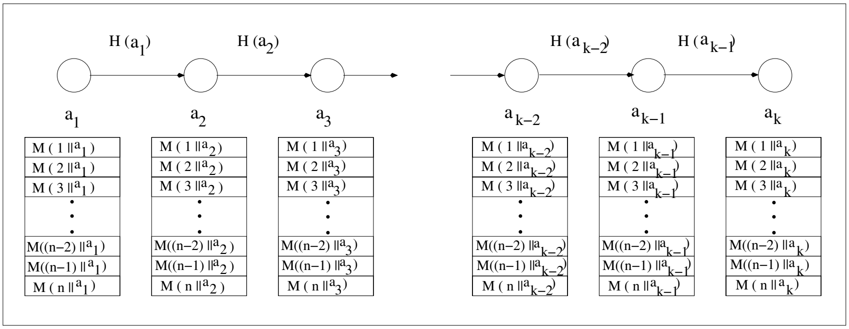


Fig. 3. Cryptographic Hash Table-Chain (HTC)

4.2 NUP-DVR Protocol

Consider the NUP-DVR protocol operation in Group 2 in Fig. 1. Let U_{24}, U_{26}, U_{27} and H_{24}, H_{26}, H_{27} be the DVR updates and the current mutually agreed pairwise hash tables, from node 2 to nodes 4, 6, 7 respectively. In Round 1, node 2 sends an update to node 4, consisting of all the updates (symmetric) encrypted with their respective hash tables (Eqn. 3). In Round 2, node 4 forwards these updates to its neighbors appropriately (Eqns. 4, 5). The symbols S, R, V and Seq represent the sender, receiver, verifiers and a sequence number (to prevent replay attacks) respectively. To avoid the update message size explosion problem, we additionally propose a novel Tree Rotation (TRot) technique employing simple checksum computations, to limit the growth of the DVR update message sizes.

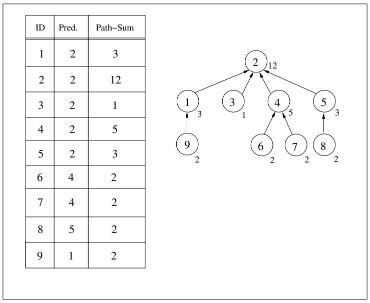
$$M1_{24} = U_{24} \oplus H_{24} || U_{26} \oplus H_{26} || U_{27} \oplus H_{27} || S_2 || R_4 || V_6 || V_7 || Seq_i \tag{3}$$

$$M2_{46} = U_{24} \oplus H_{46} || U_{26} \oplus H_{26} || S_2 || R_4 || V_6 || V_7 || Seq_i \tag{4}$$

$$M2_{47} = U_{24} \oplus H_{47} || U_{27} \oplus H_{27} || S_2 || R_4 || V_6 || V_7 || Seq_i \tag{5}$$

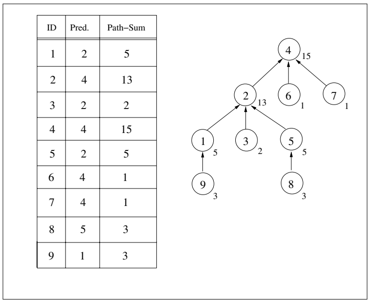
4.3 Tree Rotation (TRot)

As explained in [5], every DVR update can be viewed as a DVR tree. This DVR tree is constructed as follows: place the sender at the root, then place the routers for which the sender is the predecessor as its children at depth 1. For every router, place it as a child node of its predecessor in the DVR tree. The pathsum metric is defined for every tree node [5], as the sum of its depth in the tree and the pathsum metrics of its immediate children. In Fig. 4, pathsum for nodes 9 and 4 are 2 and 5 respectively. We now formulate Tree Rotation (TRot) based on this pathsum property. Consider a sample DVR update (and its corresponding DVR tree) sent by node 2, for the topology in Fig. 1, as shown in Fig.4. If we now re-orient the tree with node 4 as its root, we get a new DVR tree and hence a new update as shown in Fig. 5. We see that the two routing updates are exactly the same because they both have the same physical underlying connectivity. Thus the DVR tree can be rotated multiple times, each having a different root, and they all would represent the same original DVR tree.



ID	Pred.	Path–Sum
1	2	3
2	2	12
3	2	1
4	2	5
5	2	3
6	4	2
7	4	2
8	5	2
9	1	2

Fig. 4. Original Distance Vector Tree



ID	Pred.	Path–Sum
1	2	5
2	4	13
3	2	2
4	4	15
5	2	5
6	4	1
7	4	1
8	5	3
9	1	3

Fig. 5. Rotated Distance Vector Tree

Interestingly, the pathsum values differ in the different DVR trees and we use this feature to provide us the needed security guarantees. To incorporate this feature into the NUP-DVR protocol, we replace the different updates U_{24}, U_{26}, U_{27} with checksums C_{24}, C_{26}, C_{27} respectively, which are calculated as in Eqn. 6. Node 2 rotates the DVR update tree for U_{24}, randomly to some root R in the tree and computes its pathsum metric, R_{ps}, one each for the nodes 6 and 7. The choice of R is unknown to the receiver and serves as the basis of the security of the proposed optimization to the NUP-DVR protocol.

$$C_{24} = U_{24}, \quad C_{26} = R^i || R^i_{ps}, \quad C_{27} = R^j || R^j_{ps} \tag{6}$$

5 Performance and Security Analysis

We compare the NUP-DVR protocol with RIP [13]. In NUP-DVR, the overhead is due to the Round 2 update messages sent to all the verifiers. Thus the number of messages sent on the network is D-fold, where D is the average node degree in the network. Also, as the routers cache the routing tables of all the neighbors, the protocol require D-fold more storage and CPU cycles. As the routing tables used for packet forwarding are updated using Round 1 messages, there is no additional latency involved in the operation of NUP-DVR protocol, and its convergence properties are similar to that of RIP. The loss of Round 1 messages can be handled as in RIP. The Round 2 messages are critical to ensure correctness, and as their loss or explicit dropping would be interpreted as malicious behavior, it would need a retransmission/acknowledgment mechanism as in TCP, to ensure proper, ordered and timely delivery of these messages.

The proposed secure DVR protocol is single-router hijack resistant, as it can detect a single malicious sender, receiver or verifier in any group. However, if two subverted routers share a direct link, the round one messages in that group can be compromised without detection. Additionally, if subverted routers in disjoint groups act in collusion, they could falsely claim a direct link between them by sharing their private keys. A simple solution to the hidden false link problem would be to extend the group concept to depth k clusters. It is to be noted that, to the best of our knowledge, no scheme provides any guarantees against single router compromise, leave alone multiple router or collusion attack scenarios.

Additionally, the problem of verifier discovery by the sender needs to be addressed. This can be easily inferred from the update that the receiver sends to the sender in the corresponding group or more simply by explicit notifications. The sender can validate the verifiers by a simple key exchange protocol (e.g. Diffie-Hellman), and then generate a mutually agreed HTC.

6 Conclusions

It has become imperative to design secure and robust routing protocols in today's Internet, that can operate in a fairly robust manner in the presence of multiple malicious routers. Using novel concepts such as Neighbor Update Propagation

(NUP), Hash Table-Chain (HTC), and Tree Rotation (TRot), we have proposed a secure DVR protocol that is *single-router hijack resistant* and provides limited protection from multi-router collusion attacks. Possible extensions include use of appropriate data anonymizing techniques along with the proposed data morphing techniques to extend these concepts to the path vector protocols (BGP), to embed confidentiality and other policies practiced by network operators.

References

1. S. Bellovin, "Security Problems in the TCP/IP Suite", ACM CCR, pp. 32-48, 1989
2. "NANOG Archives(wow, AS7007!)", http://www.merit.edu/mail.archives/nanog/
3. "NANOG Archives(C&W Routing Instability)", http://www.merit.edu/mail.archives/nanog/
4. Smith, Murthy, Garcia-Luna-Aceves,"Securing Distance Vector Routing Protocols",SNDSS 1997
5. A. Chakrabarti, G. Manimaran, "An Efficient Algorithm for Malicious Update Detection & Detection in Distance Vector Protocols", IEEE ICC 2003
6. K. Bhargavan, D. Obradovic, C. Gunter. "Formal Verification of Standards for Distance Vector Routing Protocols", J. ACM, 49(4): 538-576, 2002
7. K. Zhang, "Efficient Protocols for Signing Routing Messages", NDSS, 1998
8. K. Bradley et. al., "Detecting Disruptive Routers: A Distributed Network Monitoring Approach", IEEE Symp. on Security & Privacy, 1998
9. Wan, Kranakis, Oorschot, "S-RIP: A Secure Distance Vector Routing Protocol", ACNS 2004
10. Y. Hu, D. Johnson, A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless AdHoc Networks", IEEE WMCSA 2002
11. Y. Hu, A. Perrig, Johnson, "Efficient Security Mechanisms for Routing Protocols", NDSS 2003
12. I. Avramopoulos, H. Kobayashi, R. Wang, A. Krishnamurthy, "Highly Secure and Efficient Routing", IEEE INFOCOM 2004
13. C. Hendrik, "Routing Information Protocol", RFC 1058, June 1988

Entropy Based Flow Aggregation

Yan Hu, Dah-Ming Chiu, and John C.S. Lui

The Chinese University of Hong Kong
yhu4@ie.cuhk.edu.hk, dmchiu@ie.cuhk.edu.hk, cslui@cse.cuhk.edu.hk

Abstract. Flow measurement evolved into the primary method for measuring the composition of Internet traffic. Cisco's NetFlow is a widely deployed flow measurement solution that uses a configurable static sampling rate to control processor and memory usage on the router and the amount of reporting flow records generated. But during flooding attacks the memory and network bandwidth consumed by flow records can increase beyond what is available. In this paper, we propose an entropy based flow aggregation algorithm, which not only alleviates the problem in memory and export bandwidth, but also maximizes the accuracy of legitimate flows. Relying on information-theoretic techniques, the algorithm efficiently identifies the clusters of attack flows in real time and aggregates those large number of short attack flows to a few metaflows. Finally, we evaluate our system using real trace files from the Internet.

1 Introduction

Traffic measurement and monitoring are crucial to operating IP networks. Especially, flow-level measurement, such as done in Cisco's NetFlow [1], is widely used for applications such as network planing, traffic profiling, usage-based accounting and security analysis. The ever increasing speeds of transmission links and high volume of traffic present great challenges for flow measurement. For high speed interfaces, the processor and the flow memory of the router can not keep up with the high packet rate. Another problem is that the volume of complete measurements of all traffic requires too much resource, both in the bandwidth required to transmit the flow records to the collector, and the resource needed to store and process the records at the collector.

A standard solution to these problems is to perform packet sampling. Cisco's sampled NetFlow uses a static sampling rate set manually according to the normal traffic volume. But when there is an anomaly such as flooding attacks in the network, the large number of small flows generated may overwhelm the router memory and the export bandwidth to the collector. One countermeasure to this problem is performing adaptive sampling, as is done in Adaptive NetFlow [2]. This algorithm guarantees a stable flow cache and export bandwidth even under severe DoS attacks. But its sampling rate could decrease to a very low level, resulting in poor overall accuracy in per flow counting including legitimate flows. Besides sampling, another method of data reduction is to do flow aggregation. Cisco implements router-based flow aggregation, which summarizes NetFlow data on the router before the data is exported to the collector.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1204–1209, 2006.

Adaptive flow aggregation [3] has recently been proposed to allow the flow monitoring systems to cope with sudden increases in the number of flows caused by security attacks. Flows of security attacks usually have some common patterns and form conspicuous traffic clusters. The algorithm identifies these traffic clusters in real-time and aggregates these large number of short flows into a few metaflows. Compared to adaptive sampling, this solution not only alleviates the problem in memory and export bandwidth, but also guarantees the accuracy of other legitimate flows. Without any predefined schemes or rules, identifying appropriate clusters and performing aggregation in real-time are not simple tasks. In this paper, we propose an entropy based flow aggregation algorithm. Based on the concept of entropy from information theory, we use the parameter of APP to indicate the priority of clusters to be aggregated. An efficient algorithm is used to identify those clusters as well as pick out some large normal flows belonging to the identified clusters.

2 Entropy Based Flow Aggregation Algorithm

We first provide a short description of our flow monitoring system, and more details can be found in [3]. The system collects network traffic data or just reads trace file and emits it as NetFlow flow records towards the specified collector, just as Cisco's NetFlow does. When the memory usage reaches a maximum value that the system allows, the system will perform flow aggregation. Using a new data structure called two-dimensional hash table, all flows with the same srcIP or dstIP will be put in one list. We also maintain a top list for srcIP and dstIP, which records the IP addresses with the most number of flows. The objectives of the old adaptive flow aggregation algorithm in [3] are, first, flow entries freed during aggregating these clusters can satisfy the memory's requirement, second, the level of these identified clusters should be as high as possible. After the algorithm identifies the desired clusters, the system merges all flows in one cluster to one metaflow. In the rest part of this section, we will describe the newly proposed entropy based flow aggregation algorithm.

2.1 Aggregation Priority Parameter (APP)

We define a *cluster* as a set of flows with the same values in one or several of the four keys, srcIP, dstIP, srcPort (plus protocol), dstPort (plus protocol), which are typically used to define a flow. We focus on clusters with a fixed srcIP/dstIP because almost all abnormal traffic has either a fixed source or destination IP address. For example, packets of DoS attacks often have the same dstIP, while packets of worm spreading usually have the same srcIP. In addition, some attacks have other fixed keys. For example, in Figure 1, all flows from one host form cluster A, while worm spreading flows from this host form cluster B. We define the biggest cluster which only has the fixed srcIP/destIP $L1$ (level 1) cluster such as cluster A, define the clusters which have fixed value in two (three) dimensions

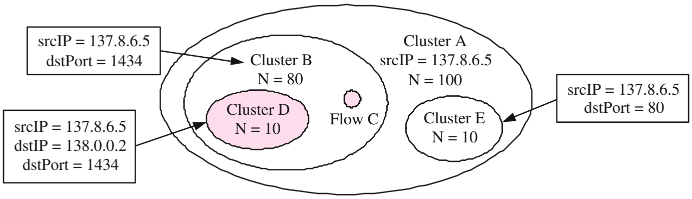


Fig. 1. Examples of clusters

$L2$ ($L3$) cluster such as cluster B (D). If we choose the higher level cluster B instead of cluster A to do aggregation, we can keep more information (srcIP and dstPort).

Besides fixed values in one or several keys, other properties of the clusters containing attack traffic include: first, the number of flows in the clusters is usually large enough to become a flooding attack; second, the size of the flows (number of packets or bytes) is often much smaller than normal flows; third, some keys other than the fixed value, such as srcIP in DoS attack traffic, dstIP in worm spreading traffic and dstPort in port scan traffic, are often randomly or uniformly distributed. In addition, if there are several big flows in the identified cluster, we would pick them out from the identified cluster and do aggregation on the rest flows, because the big flows may be normal flows mixed with attack flows. Then now the concept of the cluster is extended to the remaining flows in the original cluster. For example, in Figure 1, large flow C and $L3$ cluster D are picked out from $L2$ cluster B, the remaining flows in cluster B can also be considered as a cluster $F := B - C - D$.

We call those dimensions which have more than one value (e.g. dstIP and srcPort of cluster B) as *random dimensions*, and those dimensions which have one fixed value (e.g. srcIP and dstPort of cluster B) as *fixed dimensions*. When all flows in a cluster are merged to one metaflow, the information of its fixed dimensions will be kept, while the information of its random dimensions will be lost. Intuitively, among all clusters in Figure 1, we should choose cluster B to do aggregation for the following reasons. First, cluster B contains enough flows compared with cluster D. Second, the degree of randomness of its random dimensions is large compared with cluster A. Third, cluster B contains one more dimension of information (dstPort) than cluster A. After picking out the big flow C and $L3$ cluster D from cluster B, the one we finally choose to do aggregation is cluster F. To characterize those properties of cluster F, we propose a metric named Aggregation Priority Parameter (APP) based on the concept of entropy.

Let random variable X be one of the four dimensions (srcIP, dstIP, srcPort and dstPort). The probability distribution on X is given by $p\ (x_i) = m_i/m$, where m is the total number of traffic observed, and m_i is the number of traffic that take the value x_i. We calculate number of traffic in terms of bytes instead of flows because we need to differentiate between big flows and small flows. Entropy of one dimension is a good indicator of its degree of uncertainty or randomness. It tells us if there are some significant values that stand out from others or all values are uniformly distributed.

APP of a cluster is defined as the minimum of the entropy of its random dimensions. The larger the APP of a cluster, the higher priority this cluster would be aggregated because it characterizes those properties we want. Firstly, high APP means the number of flows in this cluster is large. Secondly, APP being large means none of those random dimensions has any significant value. In Figure 1, APP of cluster A is small than cluster B because it has a significant value 1434 in the dimension of dstPort. Third, the cluster has no flow much larger than other flows because we compute entropy in terms of number of bytes.

2.2 Algorithm Description

Using the data structure and top list in our flow monitoring system, now we have some big $L1$ clusters with fixed srcIP or dstIP. What our entropy based flow aggregation algorithm should do is that, for every $L1$ cluster, find out its sub-clusters which have the largest APP. These identified sub-clusters could not be subordinative to or overlap with each other. Among them, the cluster whose APP is the largest will be chosen. However, if several sub-clusters do not contain or overlap with each other (we call them *distinct cluster*) and have similar APP, they would all be identified.

Algorithm 1. finding out sub-clusters

```
Input:   Cluster C; random dimensions: RD
Output:  sub-clusters of high $APP$: CList
FindSubCluster(C, RD) {
  for every dimension d in RD
     $C_m[d]$ = GetMaxEntropySubset (dimension d of cluster C);
  end for
  $C_P$ = MaxAPPCluster (C, $C_m[d]$);
  for every dimension d in RD
     for every $S_i$ whose number of flows greater than $f_r$
        CList[d] = CList[d] + FindSubCluster($S_i$, RD-d);
     end for
  end for
  CList = MaxAPPDistinctCluster ($C_P$, CList[d]);
  return CList;
}
```

We use Algorithm 1 to get the sub-clusters with the largest APP. The input to the function is a cluster C with random dimensions RD. The output of the function is a list of its sub-clusters with the largest APP. First, for each random dimension d of cluster C, the function finds out its maximum entropy subset

$C_m[d]$. Maximum entropy subset is a subset of a cluster with the maximum entropy among all subsets of this cluster. For example, we assume all flows in cluster B have the same size except flow C, whose size is 10 times of that of other flows. Cluster D has 10 flows with the same dstIP. Then for the dimension of dstIP, the entropy of cluster B is 5.73, while the entropy of cluster F is 6.11, which is the maximum entropy of all subsets of cluster B. We use an efficient algorithm to find the maximum entropy subset of dimension d of a cluster C. For more details, please refer to technical report version of this paper [4].

Cluster C and $C_m[d]$ are not distinct clusters, the one with the largest APP (C_P) is chosen as a candidate for the desired sub-clusters. The fact is there may be some sub-clusters other than those maximum entropy subsets that have larger APP. They may be picked out because their sizes are large enough, or their sizes may be so small that they are subsumed in the maximum entropy subsets. So we need to recheck those sub-clusters (S_i) whose number of flows is large enough to have a large APP, as described in line 5 to 9 in the function. The last step as stated in line 10 is to choose several distinct sub-clusters from these candidates including C_P and $CList[d]$.

After the algorithm identifies the desired clusters, the system merges all flows in one cluster to one metaflow. The number of packets/bytes is the sum of packets/bytes of all aggregated flows. When new incoming packets do not belong to any active flow but belong to one metaflow, the number of packets/bytes of this metaflow will be updated. So we can get accurate packet and byte counts for the metaflow. The number of flows of the metaflow can not be counted directly. We use the multiresolution bitmap algorithm proposed in [5] to estimate it.

3 Experimental Evaluation

In this section, we use experiments to evaluate our *entropy based flow aggregation* algorithm, and compare its performance with *adaptive flow aggregation* algorithm in [3] and *adaptive NetFlow* in [2]. Under normal conditions, our system works just as basic NetFlow does. When the memory usage exceeds a predefined maximum memory, our system will perform flow aggregation, while *adaptive NetFlow* will decrease the sampling rate. Without memory constraint, *basic Netflow* can get accurate result for any flow aggregate. We use *basic Netflow* as the benchmark, and compare the performance of the three solutions. The data set we use is a 5 minute trace of the traffic on an OC48 IP backbone link, provided by Caida. We artificially generate a "DDoS" data set which simulates a DDoS attack on a single victim, and mix it with the OC48 data set.

The comparison on memory usage, export bandwidth, CPU run time and more details about the experiment can be found in [4]. Here we give out some examples of the relative error of these three schemes, as shown in Table 1. These hosts are chosen from the top dstIPs, and the top 1 is the victim of the DDoS attack. For the number of bytes, both *adaptive flow aggregation* and *entropy based flow aggregation* give out accurate results for all these hosts, while *adaptive NetFlow* affects the accuracy inevitably. Some hosts also have accurate results for

Table 1. Relative error (%) of destination IP address breakdown

dstIP	% of total	adaptive NetFlow		flow aggregation		entropy-based	
		byte Err.	flow Err.	byte Err.	flow Err.	byte Err.	flow Err.
162.131.189.129	30.7	0.83	81.14	0.00	12.88	0.00	12.99
162.131.199.254	12.4	0.41	41.53	0.00	0.00	0.00	0.00
162.131.175.235	9.5	0.66	56.18	0.00	0.00	0.00	0.00
241.46.185.161	3.2	0.57	37.23	0.00	0.78	0.00	0.00
241.46.188.127	2.6	0.49	46.65	0.00	0.16	0.00	0.00
0.3.117.37	2.1	2.02	48.17	0.00	0.15	0.00	0.00

the number of flows, which are not affected by the flow aggregation because they do not belong to the identified clusters. The new entropy-based flow aggregation algorithm accurately identifies the cluster of DDoS attack, so only the flow counter to the victim host is affected. However, the old algorithm identifies and aggregates some other clusters (eg. web traffic to host 241.46.185.161), so flow errors of those hosts do not equal to 0.

4 Conclusion

To overcome NetFlow's problem of overrunning available memory for flow records during abnormal situations, this paper proposes an entropy based flow aggregation algorithm. Based on the concept of entropy from information theory, we use the parameter of APP to indicate the priority of clusters to be aggregated. The algorithm can efficiently identify the clusters containing attack flows as well as pick out some large normal flows belonging to the identified clusters. After identifying these clusters, the system merges flows in the clusters to metaflows, and updates information of the metaflows from new incoming flows belonging to these clusters. The measurements for bytes and packets for the metaflows are completely accurate, and measurements for flows are nearly accurate using the bitmap algorithm. We use experiments on real trace file to evaluate our system and compare it with *adaptive NetFlow* and *adaptive flow aggregation*. The results show that our solution provides better accuracy.

References

1. http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml.
2. Estan, C., Keys, K., Moore, D., Varghese, G.: Building a better netflow. In: Proc. SIGCOMM '04. (2004)
3. Hu, Y., Chiu, D.M., Lui, J.: Adaptive flow aggregation - a new solution for robust flow monitoring under security attacks. In: Proc. NOMS '06. (2006)
4. Hu, Y., Chiu, D.M., Lui, J.: Entropy based flow aggregation: Tech. report (2006) http://personal.ie.cuhk.edu.hk/~yhu4/paper/entropy_tech.pdf.
5. Estan, C., Varghese, G., Fisk, M.: Bitmap algorithms for counting active flows on high speed links. In: Proc. IMC '03. (2003)

Monitoring Wireless Sensor Networks Using a Model-Aided Approach

Chongqing Zhang, Minglu Li, Min-You Wu, and Wenzhe Zhang

Department of Computer Science and Engineering,
Shanghai Jiaotong University, Shanghai, China
zhangchongqing@sjtu.edu.cn

Abstract. A wireless sensor network may consist of a large number of small, battery-powered, wireless sensor nodes and works in an unattended way. In order to manage the sensor network and collect data from the network efficiently, we need to know the state of the WSN. In this paper, we propose a model-aided approach to support the monitoring of the state of WSNs. In this approach, models are created on base station to support the monitoring of the network, and mobile agents are injected into the network to collect state information. Experimental results show the effectiveness of our approach.

1 Introduction

A WSN may consist of a large number of low-power sensors and/or actuators with limited sensing, processing, and wireless communication capabilities. After being deployed, those nodes self-organize into an integral network and work in an unattended way. In order to work properly and efficiently, applications need to reconfigure and adapt themselves based on the state information of the WSN.

For example, Database technology has been adopted by many works [1, 2] as an effective way for managing the data of WSNs. Users generally interact with a WSN database by queries and responses. In order to work out query plans of high efficiency, the base station needs to know overall state of the network so as to parse and optimize the queries submitted by users. Yet these works didn't discuss how to obtain the state knowledge of WSNs. Knowing the state of a WSN can also facilitate the network management work. By knowing the state of the WSN, users can get the knowledge of the health condition of the network and thus network management works, e.g. incremental deployment of sensor nodes, can be done efficiently.

In [3], Jerry Zhao, et al, proposed an approach called Sensor Network Tomography for monitoring the state of WSN. Instead of collecting detailed state information from each individual sensor node and then process centrally, their approach builds abstracted scans of sensor network health by combining local scans piecewise on their way towards a collecting point. And in [4], they implemented a residual energy scan (eScan) which approximately describes the remaining energy distribution within a WSN. By adopting aggregation techniques, the communication cost of Sensor Network Tomography can be reduced. Budhaditya Deb, et al, introduced in [5] a topology discovery algorithm for WSNs. The algorithm uses only a set of distinguished nodes to reply back to the topology

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1210 – 1215, 2006.

discovery probes. And by using the retrieved information, approximate topology of the network can be constructed.

In this paper, we proposed a novel model-based approach for monitoring state of WSNs. The main idea of this approach lies in the models that depict the state of WSNs and how the state changes. Those models take full advantage of the rules of how the state of WSNs changes, the relations or correlations between the attributes of sensor nodes or sensor groups. By using proper models to predict the state of WSNs, this approach can overcome the long delay and probe effect introduced by the approaches mentioned above; and the energy cost can also be reduced significantly.

The rest of this paper is organized as follows. In section II, We give the WSN model on which our research are based and present an overview of our approach. In section III, the WSNs state monitoring architecture and state information collecting methods are presented. Experimental results are presented in section IV to show the effectiveness of our approach. We conclude in section V.

2 WSN Model and Overview of Approach

2.1 Wireless Sensor Network Model

Without loss of generality, the WSNs model used in this paper is based on following assumptions:

1) A WSN is composed of a base station and large number of nodes scattered on a plane. Each node has a unique identifier. Nodes don't have to be homogeneous.

2) Base station and nodes can move at a relatively low speed.

3) Software environments that support mobile agents are installed on base station and sensor nodes [6].

2.2 Overview of Approach

The change of the state of a WSN follows some rules, and there are relations and correlations between different states of the WSN or sensor node. The rules and relations enable us to estimate or predict the state of the WSN by corresponding techniques.

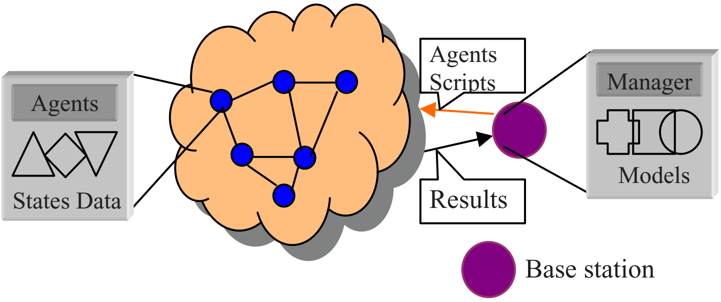


Fig. 1. Overview of Our Approach

To make full use of those rules and relations, we propose an approach for monitoring the state of WSNs. Fig. 1 gives an overview of our approach. Models depicting the state of the WSN are created on the base station, and there is a manager that is responsible for the management work of the models. The models not only can be used to support query processing and network management work, but also can be used to collect state information of the WSN from sensor nodes. The state information collecting work can be done using mobile agents. The models manager can use many strategies to collect state information of the WSN. For example, when the models can't reflect the real state of the WSN faithfully, the models manager will issue corresponding agents to collect needed state information.

3 Monitoring Architecture

Monitoring the state of a WSN is a challenging work and deserves being studied carefully. The monitoring system should introduce minimal impact on network lifetime, scale with network size; yet preserve the fidelity of the overall picture of the WSN.

3.1 Monitoring Architecture

As the reply to above-mentioned challenges, we propose two WSN state monitoring architectures to facilitate the monitoring of flat or hierarchical WSNs [7]. The architectures are illustrated by Fig.1 and Fig. 2 respectively. As for a flat WSN, on the base station side, there is a models manager in charge of the management work of the models. The models manager also issues agents with different triggering conditions and injects them into the network to collect state information from sensor nodes. On sensor nodes side, agents issued by the base station monitors the state changes of the node. When the conditions of an agent are met, it sends the state information back to the base station.

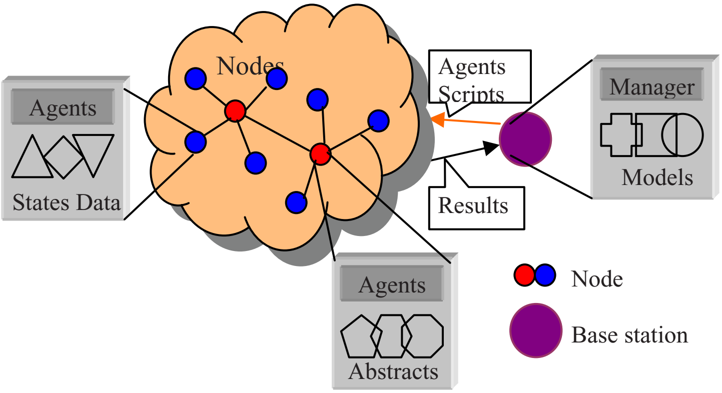


Fig. 2. State Monitoring Architecture for Hierarchical WSNs

In a hierarchical WSN, nodes are grouped into clusters and each cluster has a cluster head. As the architecture for flat WSNs, the models manager of base station also manages models and issues agents and injected them into the network, yet these agents are only issued for cluster heads. A cluster head also issues agents and sends them to the nodes belonging to the cluster. The agents that reside in the nodes belonging to the cluster send state information to the cluster head, and then the state information is aggregated into abstracts depicting the state of the cluster. And then the abstracts are reported to the base station. Note that the cluster head of a cluster may change, so agents should be able to move from old cluster head to new cluster head.

3.2 Using Agents to Collect State Information

As figure 3 depicts, the state information collecting approach has following functional steps among which steps 2, 4, 5, 6 run on base station, steps 3, 7 run on nodes, and step 1 runs on both base station and nodes.

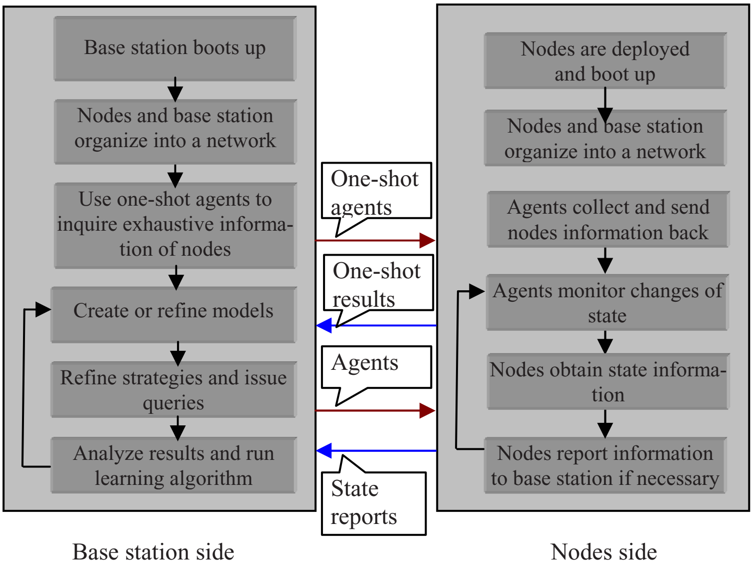

Base station boots up
Nodes and base station organize into a network
Use one-shot agents to inquire exhaustive information of nodes
Create or refine models
Refine strategies and issue queries
Analyze results and run learning algorithm
One-shot agents
One-shot results
Agents
State reports
Nodes are deployed and boot up
Nodes and base station organize into a network
Agents collect and send nodes information back
Agents monitor changes of state
Nodes obtain state information
Nodes report information to base station if necessary
Base station side
Nodes side

Fig. 3. Collecting State Information from Nodes

4 Experimental Results

As Fig. 4, one of the simulation scenes, shows, the WSN model consists of 200 sensor nodes scattered on a 300m×300m square area. The base station is located at the border of the simulation area. All nodes have same transmission ranges of 40 meters. 10 nodes can move along a straight line and at a relatively low speed less than 0.2 m/s. The initial energy of a sensor node is 5 joules, and the energy of the base station is infinite. The energy needed for a sensor node to receive and transmit a packet is

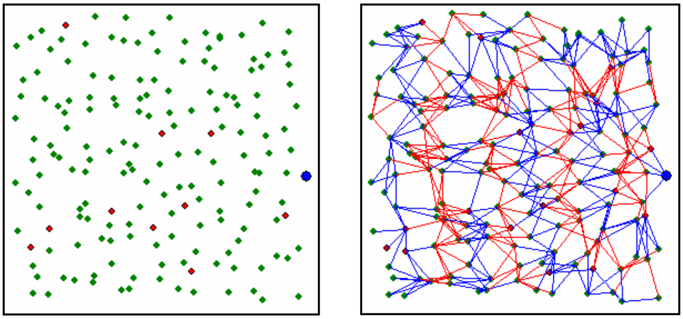


Fig. 4. An Example of WSNs Used in Experiments

2×10^{-6} joule and 1×10^{-5} joule respectively. The power for a mobile node to move is 5×10^{-5} w. For simplicity, an agent and the state information of a node are all transmitted as a data packet.

We compare five metadata management approaches: 1) NMLQ is not model-aided, and agents collect and report state information to the base station periodically; 2) NMR is not model-aided, and agents only report state information to the base station as significant changes happen; 3) MLQ is model-aided, and agents collect and report state information to the base station periodically; 4) MR is model-aided, agents only report state information to the base station as significant changes happen; 5) MOQ is model-aided, and state information is collected only when the confidence of a model is less than corresponding threshold.

We use two metrics, energy cost and fidelity to evaluate different approaches in our simulation. We simply calculate the energy cost by taking count of the data packets used for transmitting state information and agents. The higher the value is; the worse is the performance. Fidelity can be evaluated by the errors between the state value given by base station and the real state of the WSN. We use the error of the number of *packets* nodes generated and relayed to evaluate the fidelity of all approaches.

Fig. 5 compares the absolute energy cost of all approaches in one minute in detail. It can be seen that the energy cost by MOQ is less than other four approaches.

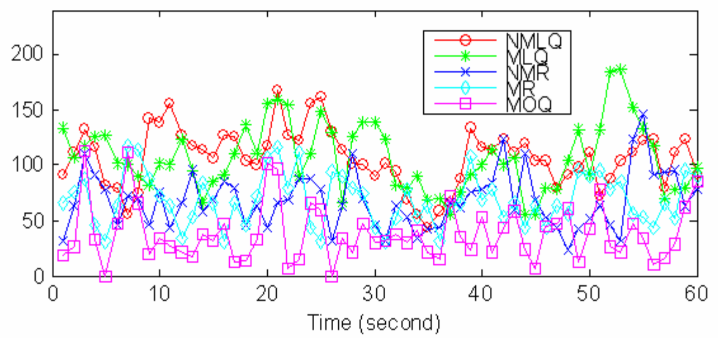


Fig. 5. Energy Consumption of All Approaches

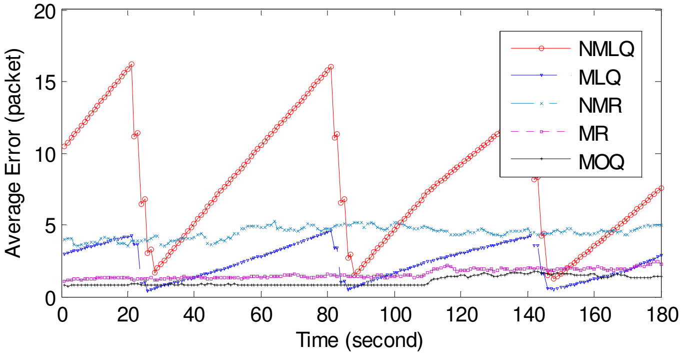


Fig. 6. Errors Comparison between All Approaches

Fig. 6 compares the errors of five approaches in 3 minutes. From the figure, the performances of model-aided MLQ and MR outscore the performances of their corresponding non-model-aided counterparts: NMLQ and NMR. Among five approaches, helped by models, MOQ consumes least energy and has the best precision.

5 Conclusion

In this paper, we propose a novel model-aided approach to support the monitoring of WSN state. This approach takes advantage of the rules of how the state of WSNs changes and relations or correlations between the attributes of nodes or nodes groups. By using proper models to predict the state of WSNs, this approach can overcome the long delay and probe effect introduced by the approaches mentioned above; and the energy cost can also be reduced significantly. Experimental results show the effectiveness of our approach.

References

1. Yao Y, Gehrke J. “The cougar approach to in-network query processing in sensor networks”. SIGMOD Record, 2002,31(3):918.
2. Sam Madden, Joe Hellerstein, and Wei Hong. “TinyDB: In-Network Query Processing in TinyOS”. Version 0.4, September 2003.
3. J. Zhao, R. Govindan, and D. Estrin. “Sensor Network Tomography: Monitoring Wireless Sensor Networks”; Student Research Poster. ACM SIGCOMM 2001.
4. J. Zhao, R. Govindan, and D. Estrin. Residual energy scans for monitoring wireless sensor networks. IEEE WCNC, 2002.
5. S. B. B. Deb and B. Nath, "A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management," Tech. rep. DCSTR-441, Dept. of Comp. Sci., Rutgers Univ., May 2002.
6. Lang Tong; Qing Zhao; Adireddy, S. Sensor networks with mobile agents. MILCOM 2003.
7. Lotfinezhad, M.; Ben Liang. Energy efficient clustering in sensor networks with mobile agents. WCNC 2005.

VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks

Peng Xie[1], Jun-Hong Cui[1], and Li Lao[2]

[1]Computer Science & Engineering Dept., University of Connecticut, CT 06029
[2]Computer Science Dept., University of California, Los Angeles, CA 90095
{xp, jcui}@engr.uconn.edu, llao@cs.ucla.edu

Abstract. In this paper, we tackle one fundamental problem in Underwater Sensor Networks (UWSNs): robust, scalable and energy efficient routing. UWSNs are significantly different from terrestrial sensor networks in the following aspects: low bandwidth, high latency, node float mobility (resulting in high network dynamics), high error probability, and 3-dimensional space. These new features bring many challenges to the network protocol design of UWSNs. In this paper, we propose a novel routing protocol, called vector-based forwarding (VBF), to provide robust, scalable and energy efficient routing. VBF is essentially a position-based routing approach: nodes close to the "vector" from the source to the destination will forward the message. In this way, only a small fraction of the nodes are involved in routing. VBF also adopts a localized and distributed self-adaptation algorithm which allows nodes to weigh the benefit of forwarding packets and thus reduce energy consumption by discarding the low benefit packets. Through simulation experiments, we show the promising performance of VBF.

1 Introduction

Recently, sensor networks have emerged as a very powerful technique for many applications, including monitoring, measurement, surveillance and control. The idea of applying sensor networks in underwater environments (i.e., forming underwater sensor networks (UWSNs)) has been advocated by many researchers [1, 4, 2]. Even though underwater sensor networks (UWSNs) share some common properties with terrestrial sensor networks, such as dense deployment and limited energy, UWSNs are significantly different from terrestrial sensor networks in many aspects: low bandwidth, high latency, node float mobility (resulting in high network dynamics), high error probability, and 3-dimensional space [2]. These new features bring many challenges to the protocol design of UWSNs. In this paper, we tackle one fundamental problem in UWSNs: robust, scalable and energy efficient routing.

Routing Challenges in UWSNs. Same as in terrestrial sensor networks, saving energy is a major concern in UWSNs. At the same time, UWSN routing should be able to handle node mobility. This requirement makes most existing

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1216–1221, 2006.

energy-efficient routing protocols unsuitable for UWSNs. Most routing protocols proposed for terrestrial sensor networks are mainly designed for stationary networks or networks with limited mobility of the sinks. They usually employ query flooding as a powerful method to discover data delivery paths. In UWSNs, however, most sensor nodes are mobile, and the network topology changes very rapidly even with small displacements due to strong multipath. The frequent maintenance and recovery of forwarding paths is very expensive in high dynamic networks, and even more expensive in dense 3-dimensional UWSNs. Thus, to provide scalable and efficient routing in UWSNs, we have to seek for new solutions. In this paper, we investigate this challenging routing problem in UWSNs, with scalability and energy efficiency as the design objectives. Moreover, robustness is also an important concern due to the high node failure rate and error-prone channels in UWSNs.

Contributions. In this paper, we propose a novel routing protocol, called vector-based forwarding (VBF), to address the routing problem in UWSNs. VBF is robust, scalable and energy efficient. It is essentially a location-based routing approach. No state information is required on the sensor nodes and only a small fraction of the nodes are involved in routing. Moreover, in VBF, packets are forwarded along redundant and interleaved paths from a source to a destination, thus VBF is robust against packet loss and node failure. Further, we develop a localized and distributed self-adaptation algorithm to enhance the performance of VBF. The self-adaptation algorithm allows nodes to weigh the benefit of forwarding packets and thus reduce energy consumption by discarding low benefit packets. We evaluate the performance of VBF through extensive simulations. Our experiment results show that for networks with small or medium node mobility (1 m/s-3 m/s), VBF can effectively achieve the goals of robustness, energy efficiency, and high success of data delivery.

2 Vector-Based Forwarding Protocol (VBF)

2.1 Overview of VBF

Vector-Based Forwarding (VBF) protocol addresses the node mobility issue in a scalable and energy-efficient way. In VBF, each packet carries the positions of the sender, the target and the forwarder (i.e., the node which forwards this packet). The forwarding path is specified by the routing vector from the sender to the target. Upon receiving a packet, a node computes its relative position to the forwarder by measuring its distance to the forwarder and the angle of arrival (AOA) of the signal[1]. Recursively, all the nodes receiving the packet compute

[1] We assume that sensor nodes in UWSNs are armed with some devices that can measure the distance and the angle of arrival (AOA) of the signal. This assumption is justified by the fact that acoustic directional antennae are of much smaller size than RF directional antennae due to the extremely small wavelength of sound. Moreover, underwater sensor nodes are usually larger than land-based sensors, and they have room for such devices [8].

their positions. If a node determines that it is close to the routing vector enough (e.g., less than a predefined distance threshold), it puts its own computed position in the packet and continues forwarding the packet; otherwise, it simply discards the packet. Therefore, the forwarding path is virtually a routing "pipe" from the source to the target: the sensor nodes inside this pipe are eligible for packet forwarding, and those outside the pipe do not forward.

2.2 The Basic VBF Protocol

In VBF, each packet carries positions of the sender, the target and the forwarder in three fields, represented by **SP**, **TP** and **FP** respectively. In order to handle node mobility, each packet contains a **RANGE** field. When a packet reaches the area specified by its TP, this packet is flooded in an area controlled by the **RANGE** field. The routing pipe is define by the vector from the sender (with position SP) to the target (with position TP) and the radius of the pipe is defined in the **RADIUS** field. Routing in VBF is initiated by query packets. VBF routes different queries in different ways:

(1) Sink_Initiated Query. There are two types of such queries: one is location-dependent query in which the sink is interested in some specific area and knows the location of the area; another is location-independent query in which the sink wants to know some specific type of data regardless of its location. For a location-dependent query, the sink issues an **INTEREST** query packet, which carries the coordinates of the sink and the target in the sink-based coordinate system, i.e., it has the information of SP and TP. This query is then directed to the targeted area following the pipe defined by SP and TP. For a location-independent query, the TP field of the INTEREST packet is invalid, and this query will be *flooded* to the target nodes. Upon receiving such query, the intended nodes can compute their locations in the sink-based coordinate system and then direct the subsequent data packets to the sink.

(2) Source_Initiated Query. If a source initiates a transmission, it first sets up a coordinate system originated at itself and floods **DATA_READY** packet into the network. Therefore, each node (including sink) can compute its location in the source-based coordinate system. The sink transforms the position of the source into its own coordinate system, and sends a location-dependent INTEREST packet to the source to allow the source to compute its position in the sink-based coordinate system for the subsequent communication.

2.3 The Self-adaptation Algorithm

In the basic VBF protocol, all the nodes inside the routing pipe are qualified to forward packets. This is not necessary in a dense network. To save energy, it is desirable to adjust the forwarding policy based on the local node density. Due to the mobility of the nodes in the network, it is infeasible to determine the global node density. Moreover, it is inappropriate to measure the density at the transmission ends (i.e., the sender and the target) because of the low propagation speed of acoustic signals. We propose a self-adaptation algorithm for VBF to allow each node to estimate the density in its neighborhood (based on local information) and adjust its forwarding accordingly.

Desirableness Factor. We introduce the notion of **desirableness factor** to measure the "suitableness" of a node to forward packets.

Definition 1. *Given a routing vector $\overrightarrow{S_1S_0}$, where S_1 is the source and S_0 is the sink, for forwarder F, the* **desirableness factor***, α, of a node A, is defined as $\alpha = \frac{p}{W} + \frac{(R - d \times cos\theta)}{R}$, where p is the projection of A to the routing vector $\overrightarrow{S_1S_0}$, d is the distance between node A and node F, and θ is the angle between vector $\overrightarrow{FS_0}$ and vector $\overrightarrow{FA}$. R is the transmission range and W is the radius of the "routing pipe".*

For a node, if its desirableness factor is large, then it is not desirable for this node to continue forwarding the packet. If the desirableness factor of a node is 0, then this node is on both the routing vector and the edge of the transmission range of the forwarder. We call this node as the **optimal node**, and its position as the **best position**. For any forwarder, there is at most one optimal node and one best position. If the desirableness factor of a node is close to 0, it means this node is close to the best position.

The Algorithm. When a node receives a packet, it first computes its position and determines if it is in the routing pipe. If yes, the node then holds the packet for a time interval $T_{adaptation}$ calculated as follows:

$$T_{adaptation} = \sqrt{\alpha} \times T_{delay} + \frac{R - d}{v_0}, \tag{1}$$

where T_{delay} is a pre-defined maximum delay, v_0 is the propagation speed of acoustic signals in water, i.e., 1500m/s, and d is the distance between this node and the forwarder. In the equation, the first term reflects the waiting time based on the node's desirableness factor: the more desirable (i.e., the smaller the desirableness factor), the less time to wait. The second term represents the additional time needed for all the nodes in the forwarder's transmission range to receive the acoustic signal from the forwarder. When two nodes are very close to the best position, Equation 1 can enlarge the delay time interval between these two nodes. During the delayed time period $T_{adaptation}$, if a node receives duplicate packets from n other nodes, then this node has to compute its desirableness factors relative to the original forwarder and these nodes $\alpha_0, \alpha_1, \ldots, \alpha_n$. If $min(\alpha_0, \alpha_1, \ldots, \alpha_n) < \alpha_c/2^n$, where α_c is a pre-defined initial value of desirableness factor $(0 \leq \alpha_c \leq 3)$, then this node forwards the packet; otherwise, it discards the packet. The theoretical analysis can be found in [8].

2.4 Performance Evaluation

We evaluate the performance of VBF through extensive simulations. We define three metrics to quantify the performance of VBF: success rate, energy consumption and average delay. The *success rate* is the ratio of the number of

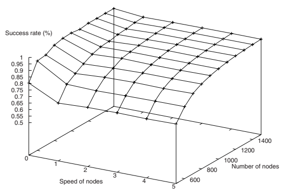


Fig. 1. Impact on success rate

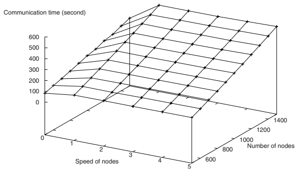


Fig. 2. Impact on comm. time

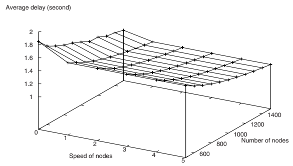


Fig. 3. Impact on average delay

packets successfully received by the sink to the number of packets generated by the source. The energy consumption is approximated by *communication time*, which is measured by the total time spent in communication, including transmission time and receiving time of all nodes in networks. The *average delay* is the average end-to-end delay for each packet received by the sink.

In our simulations, sensor nodes are deployed uniformly in a space of 100 × 100 × 100. They can move in horizontal two-dimensional space, i.e., in the X-Y plane (which is the most common mobility pattern in underwater applications). The transmission range is set to 20 meters. In order to have a bigger number of hops, the source and the sink are fixed at (90,90,100) and (10,10,0), respectively. All other nodes in the network are mobile with the same movement pattern (random walk) unless specified otherwise. The source sends data packets at the rate of 2 packets per second. The data packet size is 76 bytes and control packet is 32 bytes. The total simulation time is 200 seconds.

We first investigate the impact of node density and mobility. In this set of experiments, all the mobile nodes have the same speed. The routing pipe radius is fixed at 20 meters. We vary the mobility speed of each node from 0 m/s to 5 m/s and the number of nodes from 500 to 1500. The simulation results are plotted in the Fig. 1, Fig. 2 and Fig. 3. This set of simulation experiments have shown that in VBF, node speed has little impact on success rate, energy consumption and average delay. It demonstrates that VBF could handle node mobility very effectively.

We have also conducted simulations to show the impact of the routing pipe radius, the effectiveness of the self-adaptation algorithm, and the robustness of VBF. Due to space limit, we will not show the results in this paper. Interested readers can refer to our technical report [8].

3 Related Work and Conclusion Remarks

VBF is essentially a geographic routing protocol. To our best knowledge, VBF is the first effort to apply the geo-routing approach in underwater sensor networks. In the literature, there are many geographic routing protocols [6, 5, 3, 7, 9], in which location information of nodes is used to determine the forwarding route. Compared with VBF, these protocols assume that the location service (i.e., positioning the nodes) is available. Thus, they do not address how to position nodes in a highly dynamic network, which in fact is the foundation of the VBF protocol. Moreover, in order to save energy, VBF adopts a self-adaptation algorithm to allow nodes to weigh the benefit of forwarding packets. This idea shares some similarity with the timer-based contention algorithm in CBF protocol [3]. The major differences between these two algorithms are two-fold: (1) the timer-based contention algorithm is designed for 2-dimensional space, not for 3-dimensional UWSNs; (2) the timer-based contention algorithm can not suppress the duplicate packets from nodes close to the optimal position.

In summary, VBF is a novel protocol designed to address the routing challenges in UWSNs. It is scalable, robust and energy efficient. Through extensive simulations, we demonstrated that for networks with small or medium node mobility (1 m/s-3 m/s), VBF can effectively achieve the goals of robustness, energy efficiency, and high success of data delivery.

References

1. I. F. Akyildiz, D. Pompili, and T. Melodia. Challenges for Efficient Communication in Underwater Acoustic Sensor Networks. *ACM SIGBED Review*, Vol. 1 (1), July 2004.
2. J.-H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: Building Scalable and Distributed Underwater Wireless Sensor Networks (UWSNs) for Aquatic Applications. UCONN CSE Technical Report: UbiNet-TR05-02 (BECAT/CSE-TR-05-5), January 2005.
3. H. F*üβ*ler, J. Widmer, M. K*ä*semann, M. Mauve, and H. Hartenstein. Contention-Based Forwarding for Mobile Ad-Hoc Networks. *Elsevier's Ad-Hoc Networks*, November 2003.
4. J. Heidemann, Y. Li, A. Syed, J. Wills, and W. Ye. Underwater sensor networking: Research challenges and potential applications. *USC/ISI Technical Report ISI-TR-2005-603*, 2005.
5. M. Heissenb*ü*ttel, T. Braun, T. Bernoulli, and M. W*ä*lchi. BLR:Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks. *Elsevier's Computer Communication Journal (Special Issue)*, 2003.
6. Y. B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *ACM/Baltzer Wireless Networks*, 6(4):307–321, September 2000.
7. D. Niculescu and B. Nath. Trajectory based forwarding and its application. In *ACM International Conference on Mobile Computing and Networking (MOBICOM'03)*, September 2003.
8. P. Xie, J.-H. Cui, and L. Li. VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks. UCONN CSE Technical Report: UbiNet-TR05-03 (BECAT/CSE-TR-05-6), February 2005.
9. M. Zorzi and R. Rao. Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: multihop performance. *IEEE Trans. on Mobile Computing*, Vol. 2, Oct.-Dec. 2003.

Hybrid ARQ Scheme with Antenna Permutation for MIMO Systems in Slow Fading Channels

Jianfeng Wang, Meizhen Tu, Kan Zheng, and Wenbo Wang

School of Telecommunication Engineering,
Beijing University of Posts & Telecommunications, Beijing 100876, China
javenwang.bupt@gmail.com

Abstract. In this paper, an equivalent model for the hybrid automatic retransmission request (HARQ) multi-input multi-output (MIMO) systems with a proper combining scheme is first introduced. Based on this effective model, we present a simple technique, termed antenna permutation scheme (APS), which permutates the transmit antennas at each retransmission to improve the diversity gain from retransmissions in the slow fading environment. The theory analysis and simulation results demonstrate that the system with APS can achieve much better bit error performance.

Keywords: Hybrid ARQ, MIMO, V-BLAST, APS.

1 Introduction

Given a fixed bandwidth and power budget, an interesting approach to increase data rates is to use multiple antennas at both ends of a wireless link [1][2][3]. An attractive MIMO system to exploit this potential is the well-know Vertical Bell Labs Layered Space-Time (V-BLAST) architecture [3]. On the other hand, reliable packet data service transmission should also be provided in the future communication systems. The use of hybrid automatic retransmission request (HARQ) is intended to ensure an extremely low packet error rate [4] for the packet communication.

Some combination of the V-BLAST system with the HARQ scheme to exploit the characteristic of the MIMO systems were proposed. H.Zheng [5] gave a information theoretic analysis on the BLAST system combined with HARQ with an equivalent structure while the combining algorithm on the receiver was not described. Onggosanusi [6] presented and compared two combining schemes based on the position of the HARQ cumulative combination before or after the V-BLAST detector. We find that the HARQ-MIMO system can be modelled as the equivalent structure [5] only if the pre-combing [6] scheme is applied.

To achieve more diversity gain from retransmissions in slow fading environment, we present and analyze a simple scheme, termed antenna permutation scheme (APS), which adapts spatial diversity gain into temporal diversity gain by permuting antennas at each retransmission and improves the performance efficiently.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1222–1227, 2006.

This paper is organized as follows. Section 2 gives the brief description of HARQ-MIMO system and the equivalent structure. Based on the structure, the antenna permutation scheme is described in details in Section 3. And in Section 4 the simulation results are presented and discussed. Finally, Section 5 gives the conclusion.

Notations: Throughout this paper, matrices and vectors are set in boldface. $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^+$ and $|\cdot|$ denote transpose, conjugate transpose , Moore-Penrose pseudo-inverse and determination of the matrix, respectively.

2 HARQ-MIMO System

Here we consider a V-BLAST scheme system with N_t transmit antennas and N_r receive antennas ($N_r \geq N_t$). The received vector at the i-th transmission is written as:

$$\mathbf{y}_{(i)} = \mathbf{H}_{(i)}\mathbf{x} + \mathbf{n}_{(i)}, i = 1, 2, ..., N \tag{1}$$

where $\mathbf{x} = [x_1 \cdots x_{N_t}]^T$ is the $N_t \times 1$ transmit symbols vector. $\mathbf{H}_{(i)}$ is the $N_r \times N_t$ channel matrix contains uncorrelated complex Gaussian fading gains with unit variance at the i-th retransmission. $\mathbf{n}_{(i)} = [n_{(i)1} \cdots n_{(i)N_r}]$ represents the white Gaussian noise of variance σ^2 observed at the receive antennas. And N denotes the number of transmissions for the same packet.

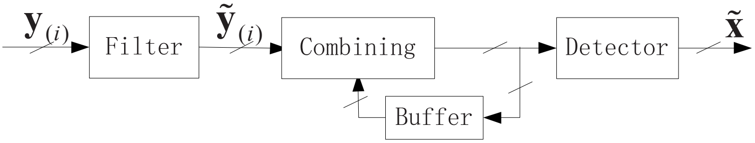


Fig. 1. Pre-combing scheme in the receiver of the HARQ-MIMO system

For the pre-combing receiver structure, as depicted in Figure.1, the received vector $\mathbf{y}_{(i)}$ and channel matrix $\mathbf{H}_{(i)}$ are first filtered with the matrix $\mathbf{H}_{(i)}^H$, yields $\tilde{\mathbf{y}}_{(i)}$, $\tilde{\mathbf{H}}_{(i)}$ respectively, i.e. $\tilde{\mathbf{y}}_{(i)} = \mathbf{H}_{(i)}^H\mathbf{y}_{(i)}$ and $\tilde{\mathbf{H}}_{(i)} = \mathbf{H}_{(i)}^H\mathbf{H}_{(i)}$ for further maximum ratio combing. Then the vectors after N transmissions are cumulatively combined. After that, some detection algorithms used in the V-BLAST system can be applied. Here the linear zero-forcing (ZF) and minimum mean square error (MMSE) detectors are considered, then we can get:

$$\tilde{\mathbf{x}}_{ZF} = (\sum_{i=1}^{N}\tilde{\mathbf{H}}_{(i)})^{-1}\sum_{i=1}^{N}\tilde{\mathbf{y}}_{(i)} \tag{2}$$

$$\tilde{\mathbf{x}}_{MMSE} = (\sum_{i=1}^{N}\tilde{\mathbf{H}}_{(i)} + \sigma^2\mathbf{I}_{N_r})^{-1}\sum_{i=1}^{N}\tilde{\mathbf{y}}_{(i)} \tag{3}$$

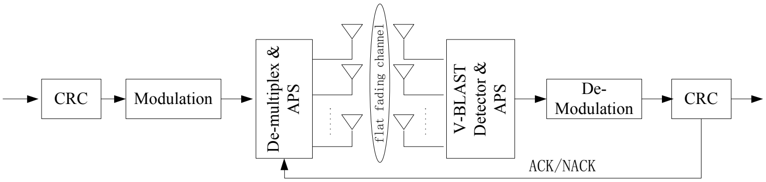


Fig. 2. HARQ-MIMO system block diagram with APS

which are not convenient to evaluate the performance directly for the cumulative sum. To facilitate our evaluation, we just adopt it to the matrix multiplication. According to the definition of $\tilde{\mathbf{y}}_{(i)}$ and $\widetilde{\mathbf{H}}_{(i)}$, the equations can be rewritten as:

$$\tilde{\mathbf{x}}_{ZF} = \hat{\mathbf{H}}^{+}\hat{\mathbf{y}} \tag{4}$$

$$\tilde{\mathbf{x}}_{MMSE} = \left(\hat{\mathbf{H}}^{H}\hat{\mathbf{H}} + \sigma^{2}\mathbf{I}_{N_r}\right)^{-1}\hat{\mathbf{H}}^{H}\hat{\mathbf{y}} \tag{5}$$

where $\hat{\mathbf{H}} = \left[\mathbf{H}_{(1)}^{T} \cdots \mathbf{H}_{(N)}^{T}\right]^{T}$ and $\hat{\mathbf{y}} = \left[\mathbf{y}_{(1)}^{T} \cdots \mathbf{y}_{(N)}^{T}\right]^{T}$. which indicate that the receiver of the HARQ-MIMO system with the pre-combining scheme can be modelled as a pure V-BLAST system with $N \times N_r$ receiver antennas, which implies that the system performance can be analyzed on this simplified equivalent structure.

3 Antenna Permutation Scheme

Because the channels that each retransmission experience are almost fully correlated, i.e. $\mathbf{H}_{(i+1)} \doteq \mathbf{H}_{(i)}$ in the slow fading environment, the temporal diversity gain from the retransmissions is limited, which is the motivation to degrade the correlation between the retransmissions. Some approaches have been considered to make the channels upon the retransmissions uncorrelated, such as [6] where a precoder matrix is used on the transmitter side. Here we present a very simple scheme which permutes the transmit antennas on each retransmission to exploit the diversity gain.

3.1 System Model

The APS-HARQ-MIMO system block diagram is depicted in Fig.2. Here the packet is sent to be modulated after CRC check-sum added, after that the main stream is de-multiplexed into multiple sub-streams, i.e. multiple transmitter antennas, in a vertical way.

At the receiver, the sub-streams from each antenna are stored in the buffer and detected based on the some V-BLAST detector algorithm. Then the soft symbol are used to determine whether the packet is error or not by CRC check.

Table 1. Antenna permutation for 2 and 4 transmit antennas

Antennas	1^{st} transmission	2^{nd} transmission	3^{rd} transmission	$\cdots$
2	$\{1,2\}$	$\{2,1\}$	$\{1,2\}$	$\cdots$
4	$\{1,2,3,4\}$	$\{3,4,1,2\}$	$\{1,2,3,4\}$	$\cdots$

If the packet is declared error-free, ACK is sent back to the transmitter, the buffer is released and the next packet is sent. Otherwise, a NACK is sent for retransmission request and some antenna permutation is selected before the next retransmission. Then combined with the previous data in buffer, the packet is checked as before.

The permutation candidates are in a pre-determined table, the 2 and 4 transmit antennas permutation strategies are illustrated in the Table 1, where the permutation strategy of the 4 transmit antennas can be optimized.

3.2 Theory Analysis

In this subsection, the performance of the system with the APS is analyzed, assumed that there are two transmissions in the $(2,2)$ system, i.e. $N = 2$, $N_t = N_r = 2$. More antennas and transmissions can by extended straightly. The channel on each retransmission is assumed to be static, i.e. $\mathbf{H}_{(i)} = \mathbf{H}$, which can be conveniently represented by a matrix $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2]$.

Here we use the linear ZF detection as an illustration. The receive signal vector is multiplied with the filter matrix $\mathbf{G}_{ZF}$:

$$\mathbf{G}_{ZF} = \hat{\mathbf{H}}^{+} \tag{6}$$

where $\hat{\mathbf{H}}$ is the equivalent channel matrix.

The estimation errors of the different layers correspond to the main diagonal elements of the error covariance matrix [7]:

$$\mathbf{\Phi}_{ZF} = E\{(\tilde{\mathbf{x}}_{ZF} - \mathbf{x})(\tilde{\mathbf{x}}_{ZF} - \mathbf{x})^H\} = \sigma_n^2(\hat{\mathbf{H}}^H\hat{\mathbf{H}})^{-1} \tag{7}$$

which equals the covariance matrix of the noise after the receive filter. So the average estimation errors from the diagonal elements is:

$$MSE = \frac{1}{2}\sigma_n^2 tr\left((\hat{\mathbf{H}}^H\hat{\mathbf{H}})^{-1}\right) \tag{8}$$

where $tr(.)$ denotes the trace of the matrix.

For the system without APS, the channels of each retransmission are static. The equivalent channel matrix can be written as $\hat{\mathbf{H}} = \left[\mathbf{H}^T\mathbf{H}^T\right]^T$. So the average error:

$$MSE_{wo} = \frac{1}{2}\sigma_n^2 tr\left(\left(\left[\mathbf{H}^H\ \mathbf{H}^H\right]\begin{bmatrix}\mathbf{H}\\ \mathbf{H}\end{bmatrix}\right)^{-1}\right) = \frac{\sigma_n^2}{4\mid\mathbf{H}\mid^2}\left(\parallel \mathbf{h}_1 \parallel_F^2 + \parallel \mathbf{h}_2 \parallel_F^2\right) \tag{9}$$

where $\parallel . \parallel_F$ denotes the Frobenius norm.

Using permutation on the transmit antennas, the channel can be written equivalently as $\mathbf{H} = [\mathbf{h}_2\ \mathbf{h}_1]$. To represent the permutation operation, a permutation matrix $\mathbf{J} = \begin{bmatrix} 0\ 1 \\ 1\ 0 \end{bmatrix}$ is introduced. Then the equivalent channel matrix can be written as $\hat{\mathbf{H}} = \left[\mathbf{H}^T\ (\mathbf{HJ})^T\right]^T$. Note that $\mathbf{JHJ}$ denotes the permutation on the main diagonal elements of the matrix $\mathbf{H}$, that is useful for the evaluations below.

Based on the equation (7), the error covariance matrix with APS can be written as:

$$MSE_w = \frac{1}{2}\sigma_n^2 tr\left(\left(\begin{bmatrix}\mathbf{H}^H\ \mathbf{JH}^H\end{bmatrix}\begin{bmatrix}\mathbf{H}\\ \mathbf{HJ}\end{bmatrix}\right)^{-1}\right) = \frac{\sigma_n^2\left(\parallel \mathbf{h}_1 \parallel_F^2 + \parallel \mathbf{h}_2 \parallel_F^2\right)}{\left(\parallel \mathbf{h}_1 \parallel_F^2 - \parallel \mathbf{h}_2 \parallel_F^2\right)^2 + 4 \mid \mathbf{H} \mid^2} \tag{10}$$

which is much less than (9) for the non-positive $\left(\parallel \mathbf{h}_1 \parallel_F^2 - \parallel \mathbf{h}_2 \parallel_F^2\right)^2$, which is related with the correlation between the transmit antennas. According to (10), we can find that the HARQ-MIMO system with APS has more diversity gain from retransmission by introducing spatial diversity into the temporal diversity in a simple way, and especially the less correlation between the spatial channel, the better performance can be achieved.

4 Numerical Results

In this section, we simulate the BER performance for the HARQ-MIMO system using QPSK modulation in the slow fading channel by simulations. The packet is transmitted twice on the channel, which is also assumed constant on each retransmission to evaluate and compare the combining gain. The APS on 2 and 4 antennas we used here is shown in the Table.1.

Within the $(2, 2)$ V-BLAST system, simulation results are presented on the left Figure.3 shows. It depicts the average BER vs. SNR for $N = 2, 4$ scenarios. The result shows that in the original HARQ-MIMO system without APS , the

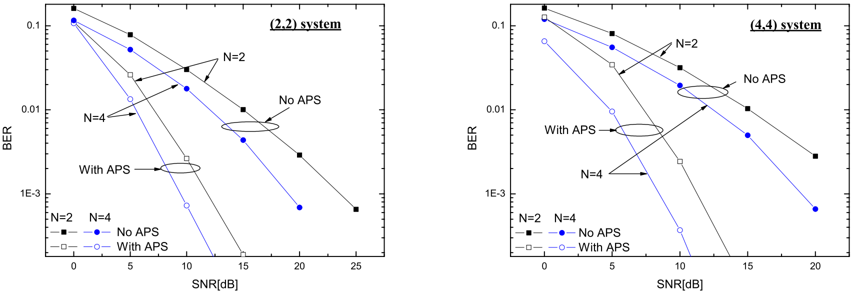


Fig. 3. Performance of the propose antenna permutation scheme in the V-BLAST system

performance is about 3dB better with the transmission number doubled, which is compliant with the analysis in the equation (9). Additionally, notice that the system with APS outperforms the original HARQ-MIMO system much better. At the BER of 10^{-2}, the performance is about 7.5dB far from the performance of the system without permutation.

The similar performance can be derived within the $(4, 4)$ V-BLAST system as the right Figure.3 shows. It shows that the system with APS outperforms the original HARQ-MIMO about 6dB at the BER of 10^{-2}. We find that the system with APS can achieve much diversity degree by a very simple scheme for the situation that un-correlation between spatial channels.

5 Conclusion

In this paper, an effective and simple scheme, termed APS, is presented for the HARQ-MIMO system in the slow fading environment. To simplify the analysis, we first introduce and explain the rationality of the equivalent structure for the V-BLAST system with pre-combining scheme. Because of the limited retransmission gain, i.e. temporal diversity gain, we permutate the transmitter antennas on each retransmission which adopt the spatial un-correlation into the temporal domain. From the theory analysis and simulation results, the proposed scheme provides significant gain with very simple implementation complexity.

References

1. E.Telatar,"Capacity of Multi-antenna Gaussian Channels," *European Transactions on Telecommunications*, vol.10, pp. 585-595, November-December 2000.
2. G.J.Foschini,"Layered space-time architechture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.,*, pp.41-59, Aut 1996
3. P.W.Wolniansky, G.J.Foschini, G.D.Golden, R.A.Valenzuela,"V-BLAST: An Architechture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel," *Proc. ITG Conf. on Source and Channel Coding*, Berlin, Germany, pp.41-59,January 2002.
4. S.Lin, D.J.Costello, M.J.Miller,"Automatic-repeat-request error-control schemes," *IEEE Communications Magazine*, vol.22, pp.5-17, December 1984.
5. H.Zheng,"Impact of Hybrid ARQ on BLAST Performance," *CISS'01*, vol.5, pp.3205-3209, May 2003.
6. E.N.Onggosanusi, A.G.Dabak, Y.H, and G.Jeong,"Hybrid ARQ Transmission and Combining for MIMO Systems," *ICC'03*, vol.5, pp.3205-3209, May 2003.
7. R.Bohnke, D.Wubben, V.Kuhn, and K.D.Kammeyer, "Reduced Complexity MMSE Detection for BLAST Architectures," *GLOBECOM '03, IEEE*, vol.4, pp.2258-2262, Dec. 2003.

Scalable Quantitative Delay Guarantee Support in DiffServ Networks Through NSIS

Jian Zhang, Maxweel Carmo, Marilia Curado,
Jorge Sá Silva, and Fernando Boavida*

Laboratory of Communications and Telematics (LCT), University of Coimbra,
CISUC-DEI, Polo II, 3030-290 Coimbra, Portugal
{zhang, maxweel, marilia, sasilva and boavida}@dei.uc.pt

Abstract. This paper investigates the issue of enabling scalable quantitative delay guarantee support in DiffServ networks through NSIS. A NSIS QoS Model is utilized to add the admission control framework to the DiffServ architecture and reservation-based admission control algorithms are designed for ingress and interior nodes respectively for enabling quantitative delay guarantees in a DiffServ domain. Due to the NSIS protocol suite can support aggregate reservations effectively and the admission control algorithms are distributed at the ingress and interior nodes, our approach can enable the quantitative end-to-end delay guarantees in a DiffServ domain while still maintaining its simplicity and scalability.

1 Introduction

The next generation Internet will provide advanced features, such as the Quality of Service (QoS) guarantees, to end-users and their applications. The DiffServ QoS architecture [2], which realizes the service differentiation by deploying a small number of pre-defined and agreed forwarding behaviors (i.e., Per-Hop Behavior (PHB)) at a DiffServ network has been paid many attentions due to its simplicity and scalability. However, DiffServ does not offer any explicit resource reservation mechanism and can only provide some level of qualitative service differentiation. Meanwhile, delay-sensitive applications, such as Voice over IP (VoIP) or Video-on-Demand (VoD), require quantitative QoS guarantees to maintain the timely arrival of their packets. Thus, to effectively support the delay-sensitive applications in a DiffServ network, extra mechanisms and algorithms, capable of offering the quantitative QoS guarantees while preserving the simplicity and scalability nature of DiffServ, must be developed. This is a challenging task for the networking researchers.

Currently, The IETF Next Steps in Signaling (NSIS) working group is working on a more generic signaling architecture than RSVP for the Internet, which consists of two distinct signaling layers: NTLP (NSIS Transport Layer Protocol) and NSLP (NSIS Signaling Layer Protocol). For the QoS signaling purpose, a

* This work has been partly supported by the European Commission under IST project EuQoS.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1228–1233, 2006.

IETF draft QoS-NSLP (NSLP for Quality-of-Service signalling) [4] has been proposed that provides a general model for each network to implement a specific QoS Model appropriate to the network technology in use and supports aggregate reservations. Hence, NSIS is an attractive signaling approach for the DiffServ.

This paper investigates the issue of enabling quantitative delay guarantee support in a DiffServ network while still maintaining its simplicity and scalability by using NSIS and distributed admission control algorithms. In particular, a NSIS QoS Model of DiffServ is utilized to signal the token bucket parameters and requested quantitative delay guarantees of incoming traffics to the edge and interior nodes of a DiffServ network, where the edge nodes maintain per-flow QoS-NSLP and reservation states whereas the interior nodes maintain only per-class states. Moreover, distributed admission control algorithms are designed for the ingress and interior nodes of the DiffServ network, respectively, based on the delay bound proposed in [3]. Since per-class aggregate reservations are fulfilled and the admission control algorithms are distributed at the ingress and interior nodes, this approach can effectively enable the quantitative delay guarantees in the DiffServ architecture while still preserving the simplicity and scalability.

The rest of the paper is organized as follows. In Section 2, we discuss related work. In Section 3, we describe our approach for scalable quantitative delay guarantee support in DiffServ networks. In Section 4, we present some concluding remarks.

2 Related Work

Provisioning quantitative QoS guarantees in DiffServ networks while maintaining the simplicity and scalability nature of DiffServ paradigm has been a challenging task for networking researchers due to the fact that the DiffServ model lacks a standard definition of signaling mechanisms and an admission control framework. A bandwidth broker, which is a central server (per domain) that arbitrates access to a statically provisioned logical partition of a network's resources, was exploited to add the admission control to DiffServ networks in [5]. All session requests are directed to the bandwidth broker, which holds a map of the network and keeps track of utilization of each resource in the network. However, due to the centralized processing nature of bandwidth brokers, a bandwidth broker has to deal with all session set up requests to its domain, which makes it exhibit the same lack of scaling (in the complexity growth sense) as IntServ.

Liao *et al.* studies the issue of provisioning quantitative differentiated services in DiffServ by proposing a set of dynamic node and core provisioning algorithms for interior nodes and core networks, respectively [6]. The node provisioning algorithm prevents transient violations of service level agreements by predicting the onset of service level violations based on a multi-class virtual queue measurement technique, and by automatically adjusting the service weights of weighted fair queueing schedulers at core routers. Persistent service level violations are reported to the core provisioning algorithm, which dimensions traffic aggregates at the network ingress edge. Note that the per-class quantitative delay guarantees provided in [6] are only in the scope of one node, i.e., the quantitative delay guarantee of a service class bounds only the packet delay across one network node. In contrast, in this paper we address the issue of provisioning end-to-end (or edge-to-edge) quantitative delay guarantees across the whole DiffServ network, which is more complex than the scenario in [6].

3 Scalable Quantitative Delay Guarantee Support in DiffServ Networks

This section first describes the network architecture and service model used in this paper and then our approach of enabling scalable quantitative delay guarantee support in DiffServ networks is presented.

3.1 Network Architecture and Service Model

We assume a DiffServ framework where all nodes in the network are NSIS aware and output-buffered, implementing class-based priority scheduling. At least 2 classes of edge-to-edge flows are considered there and we refer to one of the classes as *priority* class (e.g., Expedited Forwarding class). At each node in the DiffServ network, packets belonging to the priority class are queued in a separate priority queue which is served at strict non-preemptive priority over any other queue. Moreover, we assume that any flow i belonging to the *priority* class is shaped to conform to a leaky bucket with parameters (r_i, b_i) when it arrives at the ingress node of the DiffServ. Furthermore, flow i indicates its traffic parameters (r_i, b_i) and its QoS requirements (in this paper, i.e., the requested end-to-end delay guarantee D_i across the DiffServ) to the ingress node. Note that various alternatives (e.g. SIP, RSVP, NSIS, etc) can be used by incoming flows to express their traffic parameters and QoS requirements to the ingress node and no assumptions are made about that here. Then, NSIS signaling messages will be exchanged in the DiffServ network to first discover the data path which an incoming flow will take to pass through the network, then to check whether there are enough resources available along each link in the data path to meet the requested delay guarantee and to reserve the appropriate resources when they are available.

3.2 Distributed Admission Control Algorithms

The distributed admission control algorithms used by the ingress and interior nodes are derived here. First, we define some parameters as follows. MTU is to denote the maximum size of any packet in the network, h_{max} to denote the maximum number of hops any flow in the network could traverse, C_l to denote the capacity of line l, S_l to denote the set of all priority flows constituting the priority aggregate on link l, P_l to denote the maximum rate with which the priority traffic aggregate is injected to link l and ρ to denote the ratio of the capacity of any link in the network to be devoted to the priority class. According to [3], a bound on the worse-case end-to-end delay for the priority class traffic is $D = \frac{h_{max}}{1-(h_{max}-1)u\rho}(\Delta + u\tau)$ provided that the three inequalities hold: $\rho < \mathtt{min}_l \frac{P_l}{(P_l-C_l)(h_{max}-1)+C_l}$, $\sum_{i \in S_l} r_i \leq \rho C_l$ and $\sum_{i \in S_l} b_i \leq \tau C_l$, where $u = \mathtt{max}_l \frac{P_l-C_l}{P_l-\rho C_l}$, $\Delta = \mathtt{max}_l \frac{MTU}{C_l}$ and τ is the parameter to be set to bound the sum of the token bucket depths burst at any link relative to the link capacity. Note that, in many cases, the depth of the leaky bucket of a flow depends linearly on the rate of the flow, such that $b_i \leq \tau_0 r_i$ for every flow i and for some τ_0. In such cases, we set $\tau = \rho\tau_0$.

Given the network topology of a DiffServ domain, we can set the value for ρ according to the above inequalities and set the value for τ based on the estimated burstiness of priority class traffic. Moreover, the value of Δ and the value of u can be determined based on the traffic dynamics and the considered network topology. Note that, if there is no information about the value of P_l, it can be set as the sum of the bit rates of all incoming links to link l. Then, when a new flow (flow i) belonging to the priority class arrives, the ingress node will first check whether this flow can be accepted to its outgoing link. If the flow is accepted, it will send a NSIS signaling message to discover the data path which flow i will pass through and during the course of the path discover, the interior nodes in the data path will perform their admission control algorithm successively to check whether or not flow i can be accepted to the links it will traverse. The admission control algorithm for the interior node (including the ingress node), whose output link is link l, is designed as follows: if both $R_{curr} + r_i \leq \rho C_l$ and $B_{curr} + b_i \leq \tau C_l$ hold, flow i can be accepted to link l, otherwise, flow i is rejected, where $R_{curr} = \sum_{j \in S_l} r_j$ and $B_{curr} = \sum_{j \in S_l} b_j$. If flow i is accepted by all interior nodes in its data path and the number of hops it will traverse in the network is h, the ingress node will perform the following admission control algorithm to decide whether this flow will be accepted or rejected to the DiffServ network: if

$$D_i \geq \frac{h}{1-(h-1)u\rho}(\Delta + u\tau) \qquad (1)$$

holds, flow i is accepted, otherwise, it is rejected, where D_i is the end-to-end delay guarantee requested by flow i. Of course, if flow i is rejected by any interior node in its data path, it will be rejected to the network by the ingress node without executing the ingress node's admission control algorithm.

It can be observed from Eq. (1) that the right item of (1) is from the delay bound in [3] except that h_{max} is replaced by h due to the fact that the exact number of hops every flow will pass through can now be obtained via the NSIS signaling exchange. Since flow i is accepted by all interior nodes in its data path means that all the above inequalities has already been satisfied by the selection of ρ for the network, the packet delay d_i of flow i will be bounded by the value of $\frac{h}{1-(h-1)u\rho}(\Delta + u\tau)$, i.e., $d_i \leq \frac{h}{1-(h-1)u\rho}(\Delta + u\tau)$. Now, if flow i is accepted to the network by the ingress node, we can obtain $d_i \leq \frac{h}{1-(h-1)u\rho}(\Delta + u\tau) \leq D_i$, i.e., the requested worse-case end-to-end delay guarantee of flow i is satisfied.

3.3 NSIS QoS Model

The NSIS QoS Model for enabling quantitative delay guarantees in DiffServ networks is presented below.

First of all, the NSIS QoS Model allows external traffics to express their traffic parameters and quantitative QoS requirements (e.g. quantitative bandwidth, delay, jitter or loss guarantee) to the ingress nodes of a DiffServ domain. Secondly, the NSIS QoS Model supports two sets of admission control algorithms (measurement-based and reservation-based admission control) to satisfy the quantitative QoS requirements of accepted flows. In this paper, the reservation-based admission control algorithms are designed to illustrate the usage of the NSIS QoS Model for enabling quantitative delay guarantees in a DiffServ domain.

In particular, at a DiffServ domain where all nodes deploying this NSIS QoS Model, the edge nodes will store and maintain per-flow NTLP, QoS-NSLP and QoS Model related reservation states. The interior nodes will be NTLP stateless, which means no NTLP states need to be stored, and be either QoS-NSLP stateless (for measurement-based admission control operation), or are reduced-state nodes storing per PHB aggregated QoS-NSLP and reservation states (for reservation-based admission control operation). As described in [4], four message types: RESERVE, QUERY, RESPONSE and NOTIFY have currently defined to support the QoS singaling operations, which consist of three types of QoS-NSLP objects including the QoS specification (QSPEC) object. QSPEC object describes the actual resources that are required and depend on the QoS Model being used. The QSPEC object of the NSIS QoS Model presented here contains three fields: the QoS Description, the Per Hop Reservation Control Information (PHR Control Information) and the Per Domain Reservation Control Information (PDR Control Information). In particular, the QoS Description field contains the following parameters: <**QoS Description**> = <Token Bucket> <PHB Class> <Bandwidth> <Path Latency> <Path Jitter> <Packet Loss Ratio> <Packet Error Ratio>, where the bit formats of all above parameters conform to the bit formats specified by the QoS-NSLP QSPEC template [1]. These parameters describe the characteristics of incoming traffics and the quantitative QoS guarantees requested to the DiffServ domain.

3.4 Scalable Quantitative Delay Guarantee Support Through NSIS

Our approach for the scalable quantitative delay guarantee support in a DiffServ domain is illustrated here. First of all, we assume that a new flow (flow i) belonging to the priority class indicates its token bucket parameters (r_i and b_i) and requested end-to-end delay guarantee D_i to one ingress node of a DiffServ domain. Then, if the flow can be accepted to the outgoing link of the ingress node, it will encapsulate parameters r_i and b_i into the <Token Bucket> field of the QoS Description and send a QUERY message to discover the data path that flow i will take and to retrieve the number of hops it will pass through. Next, each interior node at the data path of flow i will execute the admission control algorithm in Section 3.2 to check whether it can be accepted to the outgoing link of the interior node. When the QUERY message reaches the egress node, the egress node will report the QUERY result to the ingress node via the RESPONSE message which will however be bypassed by those interior nodes. If the QUERY result is positive, the ingress node will perform the admission control algorithm of Eq. (1) to check whether the requested end-to-end delay guarantee D_i can be satisfied or not. If D_i can be met by the DiffServ domain, the ingress node will send the RESERVE message to reserve the pertinent resources for flow i along the data path already discovered, in this paper, i.e., each interior node (including the ingress node) will add r_i and b_i to its R_{curr} and B_{curr} of the priority class, respectively. After the ingress node receives the positive RESERVE report from the egress node, it will mark the packets of flow i as priority class (here, i.e., EF class) and admit them into the DiffServ network. Otherwise, flow i is rejected.

4 Conclusions

This paper investigates the issue of enabling scalable quantitative delay guarantee support in DiffServ networks through NSIS. A NSIS QoS Model for DiffServ networks is utilized to signal the token bucket parameters and requested quantitative delay guarantees of incoming traffics to the edge and interior nodes in a DiffServ network, where the edge nodes maintain per-flow QoS-NSLP and reservation states whereas the interior nodes maintain only per-class states. Moreover, distributed admission control algorithms are designed for the ingress and interior nodes of the DiffServ network, respectively, based on the delay bound proposed in [3]. Due to the NSIS protocol suite can support aggregate reservations effectively and the admission control algorithms of our approach are distributed at the ingress and interior nodes, our approach can enable the quantitative end-to-end delay guarantees in a DiffServ domain while still maintaining its simplicity and scalability.

References

1. J. Ash, A. Bader and C. Kappler. QoS-NSLP QSPEC Template, Internet draft, work in progress, Internet Engineering Task Force, October 2005.
2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss. An Architecture for Differentiated Service, RFC 2475, Internet Engineering Task Force, December 1998.
3. A. Charny and J.Y. L. Boudec. Delay Bounds in a Network with Aggregate Scheduling, Proc. of QofIS 2000, LNCS 1922, pp. 1-13, 2000.
4. J. Manner, G. Karagiannis, A. McDonald, S. Van den Bosch. NSLP for Quality-of-Service signalling, Internet draft, work in progress, Internet Engineering Task Force, October 2005.
5. K. Nichols, V. Jacobson and L. Zhang. A two-bit differentiated services architecture for the Internet, RFC 2638, Internet Engineering Task Force, July 1999.
6. R.-F. Liao and A. T. Campbell. Dynamic Core Provisioning for Quantitative Differentiated Services, IEEE Transactions on Networking, 2004.

SDC: A Distributed Clustering Protocol for Peer-to-Peer Networks

Yan Li[1], Li Lao[2], and Jun-Hong Cui[1]

[1] Computer Science & Engineering Dept., University of Connecticut, CT 06029
[2] Computer Science Dept., University of California, Los Angeles, CA 90095
yan.li@uconn.edu, llao@cs.ucla.edu, jcui@engr.uconn.edu

Abstract. Network clustering can facilitate data discovery and peer-lookup in peer-to-peer systems. In this paper, we design a distributed network clustering protocol, called SCM-based Distributed Clustering (SDC), for peer-to-peer networks. In this protocol, clustering is dynamically adjusted based on Scaled Coverage Measure (SCM), a practical clustering accuracy measure. By exchanging messages with neighbors, peers can dynamically join or leave a cluster so that the clustering accuracy of the whole network is improved. SDC is a fully distributed protocol which requires only neighbor information, and it can handle node dynamics locally with very small message overhead while keeping good quality of clustering. Through extensive simulations, we demonstrate that SDC can discover good quality clusters very efficiently.

1 Introduction

In a peer-to-peer system, there are usually large numbers of peers. And the knowledge of each peer about the network topology is usually limited to its immediate neighbors. Due to the large scale and the lack of knowledge about the complete network structure in each peer, a main challenge in peer-to-peer system design is to effectively perform data discovery and peer look-up. The *network clustering* technique can significantly facilitate these operations [1] [2].

Network clustering is the procedure of partitioning a network topology into groups or clusters. It can be performed in both centralized and distributed ways. Centralized network clustering is an off-line procedure, in which complete network topology information need to be obtained before clustering. In our work, we focus on the latter one. We are interested in the network clustering of large-scale peer-to-peer networks.

There are several characteristics of a good distributed clustering protocol. First of all, as a natural requirement of network clustering, nodes in the same clusters should be highly connected, and less connected between clusters. Secondly, the protocol should well control the cluster size (or cluster diameter). Thirdly, the protocol should result in a minimum number of “orphan” nodes. Lastly, a good clustering protocol should take node dynamics into account, since the target networks (peer-to-peer networks) are highly dynamic with frequent entry and exit of nodes.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1234–1239, 2006.

In the literature, there have been considerable research efforts addressing the problem of network clustering, but very few of them studied the problem of clustering in peer-to-peer networks. Among the existing approaches, MCL [5] is well accepted as an efficient and accurate network clustering algorithm. However, this approach assumes that the complete network topology is available at one central point, which is not realistic in peer-to-peer systems. CDC [4], on the other hand, is a distributed algorithm. It forms clusters based on node connectivity. The main issue with this algorithm is that it can not handle node dynamics in a decent way, as limits its utility in peer-to-peer networks.

With these problems in mind, we design a novel network clustering protocol called **SCM-based Distributed Clustering (SDC)**, which satisfies all the design criteria discussed above. In this protocol, clustering is dynamically adjusted based on Scaled Coverage Measure (SCM) [6], a practical clustering accuracy measure. By exchanging messages with neighbors, peers can dynamically join or leave a cluster so that the clustering accuracy of the whole network is improved. To control the cluster size, TTL (Time-To-Live) is piggybacked in exchange messages to guarantee the cluster diameter will never exceed a predefined threshold. SDC is a fully distributed protocol which requires only neighbor information, and it can handle node dynamics locally with very small message overhead while keeping good quality of clustering. Through extensive simulations, we demonstrate that our proposed protocol, SDC, is able to discover good quality clusters in a very efficient way.

2 Network Model and Scaled Coverage Measure

Network Model. We assume each peer-to-peer network is represented by a connected, undirected graph $G = (V, E)$, where V is the set of nodes corresponding to the set of peers in the system and E is the set of links, which are the logical connections between peers. We denote $|V| = n$ and $|E| = m$. Then the partition $\mathcal{C} = \{C_1, C_2, ..., C_l\}$ of V is called a *clustering* $\mathcal{C}$ of graph G, and C_is are called *clusters*. Each cluster should be a non-empty subset of V. Obviously, $\bigcup_{i=1}^{l} C_i = V$. The *diameter* of a cluster C_i is defined as the maximum length of the shortest paths among all pairs of nodes in C_i. Then if a cluster has only one node, it has a diameter of 0. We call the clusters with diameter 0 as *orphan nodes*. In this paper, we also define *cluster size* as the number of nodes in a cluster to represent the cluster scope. Clearly, cluster size and cluster diameter are closely related. In most context, “control cluster size” and “control cluster diameter” have the same meaning of “control cluster granularity”. We only differentiate these two concepts in the protocol description.

SCM is a practical measure to evaluate the accuracy of connectivity based clustering proposed by S.Van Dangon [5]. We assume $\mathcal{C} = \{C_1, C_2, ..., C_l\}$ is a clustering on network $G = (V, E)$. Given a node $v_i \in V$, we have the following notations: $\mathbf{Nbr}(v_i)$ is the set of neighbors of node v_i; $\mathbf{Clust}(v_i)$ is the set of nodes in the same cluster as node v_i (excluding v_i); Then, two special sets of nodes associated with v_i are defined as follows: $\mathbf{FalsePos}(v_i, \mathcal{C})$ is the set of nodes in the same cluster as v_i but not neighbors of v_i; **FalseNeg**$(v_i, \mathcal{C})$ is the set of neighbors of v_i but not in the same cluster as v_i. The SCM of node v_i is defined as follows:

$$SCM(v_i) = 1 - \frac{\mid FalsePos(v_i, \mathcal{C}) \mid + \mid FalseNeg(v_i, \mathcal{C}) \mid}{\mid Nbr(v_i) \cup Clust(v_i) \mid}. \tag{1}$$

For graph G, $SCM(G)$, is defined as the average of the SCM values of all the nodes, that is, $SCM(G) = (\sum_{v_i} SCM(v_i))/n$, which lies in $[0, 1]$.

SCM well reflects the significance of clustering features in a given network. First of all, it is easy to see that the higher the SCM, the smaller the connectivity between clusters and the higher the connectivity within clusters. For graphs containing only isolated clusters/subgraphs that are themselves fully connected, the SCM value is 1. Secondly, for any graph, there exists a highest SCM value which is determined solely by the network structure. If the network does not contain significant clustering substructures, this highest "available" SCM value can be very small. However, if we evaluate two clustering techniques on the same network, the one which results in a higher SCM value discovers more accurate clustering substructures than the one with smaller SCM value, although both resultant SCM values could be very small. Lastly, the SCM value of an orphan node is 0, which matches our goal of minimizing the number of orphan nodes.

Based on the definition of SCM, the network clustering problem can be simplified as partitioning a network topology so that its SCM is maximized. Our proposed SDC protocol exactly follows this idea, adaptively forming clusters in an aggressive manner.

3 The SDC Protocol

3.1 Protocol Description

The SDC protocol performs in a fully distributed way. Each node v_i only needs to maintain some basic information about its neighbors and the cluster it belongs to, such as the cluster id *clust_id*, the cluster size *clust_size* (which is the total number of nodes in the cluster).

Given a network, each node v_i is initialized as an orphan node with its own *clust_id* (any unique *id* is sufficient) and *clust_size* (1 in this case). And the two parameters for SCM computation b_{v_i} and a_{v_i} are initialized as $degv_i$. Then all nodes start to exchange messages with their neighbors, conduct some simple computation, and form clusters in a greedy manner. After a number of rounds of communication, the clustering procedure becomes stable without further message exchange and the network is finally clustered.

In SDC, we define a set of **Clust_** type of messages. Suppose node v_i wants to be clustered. The following clustering messages may be involved.

- **Clust_Probe.** Node v_i first sends the message *Clust_Probe* to every node $v_j \in Nbr(v_i)$ to find out other clusters in the neighborhood. Each node which receives *Clust_Probe* will send its *clust_id* and *clust_size* back to v_i.

- **Clust_Request**. Once receiving the *clust_id*s from its neighbors, node v_i can determine its "neighbor clusters". Suppose v_i discovers that a cluster Cl is connected with it, it issues a *Clust_Request* message which is flooded in Cl and v_i's current cluster $Clust(v_i)$. This is a well-controlled flooding, since upon receiving *Clust_Request*, a node can forward this message to others only if it is in Cl or $Clust(v_i)$. For any node v_j in cluster Cl, upon receiving *Clust_Reqest*, a very simple computation is performed to obtain $\Delta SCM(v_j)$, the gain in $SCM(v_j)$ assuming node v_i joins Cl. This computation only requires the information of whether v_i is v_j's neighbor or not. Similarly, for any node $v_k \in Clust(v_i)$, it needs to compute $\Delta SCM(v_k)$ as if v_i leaves its current cluster.

  To control the number of exchanged messages, a TTL is carried in *Clust_Request*. Once receiving *Clust_Request*, any node should check the TTL value first and will discard the message without forwarding to others if TTL expires. TTL is also used to control the cluster diameter.
- **Clust_Reply.** Upon receiving *Clust_Request* from v_i ($TTL \neq 0$), node v_j sends back a *Clust_Reply* message carrying $\Delta SCM(v_j)$ and v_j's *clust_id* back to node v_i.
- **Clust_Reject.** Based on the TTL in *Clust_Request*, node $v_j \in Cl$ can determine whether or not the cluster diameter will exceed the predefined threshold due to the joining of node v_i. If this is the case ($TTL = 0$), v_j simply stops forwarding *Clust_Request* to other nodes and a *Clust_Reject* message will be sent back to v_i. Once receiving *Clust_Reject*, node v_i will not join Cl.
- **Clust_Update.** After node v_i receives *Clust_Reply* messages from all the nodes in its current cluster and the neighbor cluster Cl (in the case that no *Clust_Reject* is received from Cl), it computes the overall gain $\Delta SCM(G)$ based on the received information, assuming it leaves its original cluster and joins Cl. If $\Delta SCM > 0$, v_i should join Cl. Once v_i determines which cluster to join, a *Clust_Update* message containing v_i's node id and its original *clust_id* is flooded in its original cluster and the new cluster it will join. Then, v_i and any node receiving this message will update the *clust_size* and their own SCM.

After node v_i joins the new cluster, its neighbors in the original cluster are affected and should check whether they should join other clusters, in the same way as node v_i does. The whole procedure will end if no node can join any cluster based on $\Delta SCM(G)$ and the cluster diameter control.

3.2 Handling Node Dynamics

Peer-to-peer networks are dynamic systems. With node entry and exit at arbitrary points, the network structure is changed and the existing clusters are affected. Re-do the whole clustering procedure may keep good clustering accuracy. However, it is very inefficient and the procedure may never stabilize if node entry and exit happens frequently. Therefore, designing an effective and efficient scheme to handle node dynamics is critical in peer-to-peer network clustering.

Our SDC protocol can naturally handle node dynamics. Whenever a new node v_i joins the system, it is first initialized as an orphan node and gets its own *clust_id* (any unique *id* is sufficient) and *clust_size* (which is 0). Since the network structure between node v_i and its neighbors is changed, a **Join** message carrying v_i's *clust_id* is issued by v_i to all of the neighbors so that they can update their SCM. As v_i's joining changes its neighbors' connectivity, the affected neighbor nodes should perform a new round of clustering procedure. When a node wants to leave, it sends a **Leave** message to each of its neighbors as well as every other node in its cluster through flooding so that the *clust_size* and SCM values of the affected nodes can be updated. This will also activate a new round of clustering procedure at these affected nodes. The idea behind this scheme comes from the fact that node entry and exit are localized events and only a few nodes are affected and need to be re-clustered.

It is clear that some overhead is introduced when SDC handles node dynamics. Nevertheless, this overhead is very small since only neighbors and/or the nodes in the same cluster are directly affected. In next section, we will show that SDC can achieve very good clustering accuracy while with low overhead in the presence of node dynamics. In contrast, CDC has to re-do the complete clustering procedure for any node join or leave in order to maintain good clustering accuracy, which introduces a lot of overhead.

3.3 Simulation Evaluations

In this section, we conduct simulations to evaluate the performance of SDC, comparing it with CDC, in dynamic systems.

Experiment Settings. We implement both the SDC and CDC algorithms and run them on different topologies. The configurable parameters used in the CDC scheme are carefully tuned so that we can get the best results for CDC. For implementation details, please refer to our technical report [3]. We use two metrics: *clustering accuracy* and *message overhead*. We compute the clustering accuracy using SCM, and measure the overhead in term of the number of exchange messages between peers.

Results and Analysis. In this set of experiments, we use power-law topologies. We fix the average degree as 10, and vary the topology size (i.e., the number of nodes) from 200 to 5000. We run each experiment more than 100 times so that all the results have a standard deviation of less than 0.1%. We measure the message overhead and clustering accuracy for arbitrary node join (and leave).

We first study node leaving. In SDC, when a node leaves the network, the affected nodes (its neighbors and the nodes in the same cluster) need to "re-cluster" in order to maintain good clustering accuracy. In CDC, upon a node entry or exit, the whole network has to be re-clustered. For comparison, we also run "SDC Reclustering", in which the whole topology redoes SDC clustering after each node exits the network. The results are plotted in Fig. 1 and Fig. 2. We observe that SDC can maintain a higher clustering accuracy than CDC while only much smaller overhead is introduced. Moreover, compared with

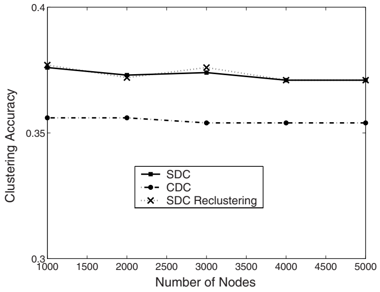


Fig. 1. Clustering accuracy on node exit

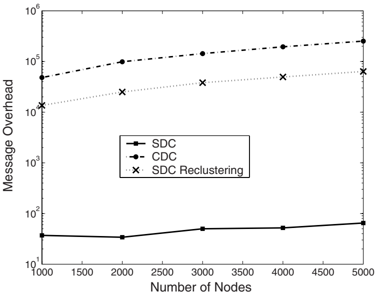


Fig. 2. Message overhead on node exit

"SDC Reclustering", SDC yields almost same accuracy values, which further demonstrates the effectiveness of SDC for node leaving. We conduct similar experiments for node joining, and obtain similar results. Thus, we conclude that SDC can handle node dynamics very effectively.

Besides the performance evaluation of SDC in dynamic systems, We also study the influence of node degree and TTL on the performance of SDC. Due to space limit, we do not show those results in this paper. Interested readers can find the complete simulation study in [3].

4 Conclusion Remarks

We have presented a distributed clustering protocol, SDC, for peer-to-peer networks. SDC can satisfy all the criteria for a good clustering algorithm: it considers node connectivity; it well-controls the cluster size; it minimizes the number of orphan nodes; and it can locally handle node dynamics with small overhead. Through simulations, we demonstrate that SDC can achieve much better performance than CDC in terms of both clustering accuracy and message overhead.

References

1. L. Garces-Erice, E. W. Biersack, K. W. Ross, P. A. Felber, and G. Urvoy-Keller. Hierarchical p2p systems. In *Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par)*, 2003.
2. G. Kwon and K. D. Ryu. An efficient peer-to-peer file sharing exploiting hierarchy and asymmetry. In *SAINT*, pages 226–233, 2003.
3. Y. Li, L. Lao, and J.-H. Cui. Sdc: A distributed clustering protocol for peer-to-peer networks. *UCONN CSE Technical Report: UbiNet-TR06-02*, February 2006.
4. L. Ramaswamy, B. Gedik, and L. Liu. A distributed approach to node clustering in decentralized peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 16(9), Sept. 2005.
5. S. van Dongen. A new cluster algorithm for graphs. *Technical report INS-R9814, Centrum voor Wiskunde en Informatica (CWI), ISSN 1386-3681*, Dec. 1998.
6. S. van Dongen. Performancde criteria for graph clustering and markov cluster experiments. *Technical report, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam*, 2000.

A New Burst Scheduling Algorithm for Edge/Core Node Combined Optical Burst Switched Networks

SeoungYoung Lee, InYong Hwang, and HongShik Park

BcN Engineering Research Center, Information and Communications University,
103-6 Munji-Dong, Yuseong-gu, Daejeon, Korea
{seoungyoung, iyhwang, hspark}@icu.ac.kr

Abstract. The burst contention problem in Optical Burst Switching network is an intrinsically serious problem. Many researches have tried to solve this problem, however it have been known that avoiding the burst loss is very difficult issues in the current OBS network. To improve burst blocking rate, we consider the edge/core combined OBS network where the core node performs the edge node function as well. Through this architecture, available amount of data burst that the node generates can be expected with respect to offset-time of transit data bursts. Any researches for this area has not been performed, thus we propose a new data scheduling algorithm for the edge/core combined OBS network where data bursts that the node generates do not interrupt transit data bursts from previous nodes. We analyzed the data burst loss rate and the throughput in relation with the offset-time of transit data bursts. Results show that the loss rate of the data bursts is drastically reduced and the throughput improves when the offset-time of transit data bursts increases.

Keywords: OBS, JET, scheduling, void filling, CoS.

1 Introduction

The emergence of wavelength division multiplexing (WDM) technology is considered as a solution to fulfill the tremendously increasing demands of transmission bandwidth driven by the growth of IP-based data traffic. At the same time, the necessity to make the next-generation optical Internet architecture is augmented, which can transport IP packets directly over the optical layer without opto-electro-optic (O/E/O) conversions, like optical packet switching (OPS). Although OPS which can achieve higher utilization is attractive, there are practical limitations such as optical buffer and all optical processing. Presently, optical burst switching (OBS) technology is under study as a solution for optical Internet backbone in the near future since OBS technology can cut through data messages without O/E/O conversion and guarantee the Class of Service (CoS) without any buffering [1-2].

The analyses of OBS have been focused on the ring network or simply mesh network. Recently, because the issues about the commercialization have been increased, the studies for the real networks, especially for the mesh-type networks, have been increased [3-4]. In this paper, to meet the research trend of OBS network, we consider the edge/core node combined (ECNC) OBS network, where all node can generate data burst (DB) with the edge node function and forward DB to the next node with the core node function. If the node control the sending time of its DB by buffering the DB

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1240 – 1245, 2006.

in the electrical buffer, it will not interrupt the transit data bursts (TDB) generated previous nodes. Therefore, the starting DB can avoid contention by inserting DB among TDB. By doing so, the overall network performance will be improved while this scheme do not affect the loss rate of TDB. We find that the offset-time of TDB affects the throughput of the network and analyze throughput mathematically. The remainder of paper has been structured as follows. Section 2 reviews the burst blocking probability in conventional OBS network. Section 3 presents the new scheduling algorithm suitable for the ECNC OBS network. Section 4 provides the analysis results, and the conclusion follows in Section 5.

2 Blocking Probability in OBS Network

2.1 Network Modeling in Conventional OBS Network

In Figure 1, the typical network model in the conventional JET OBS network is shown. In this model, the ingress node 1, 2, and 3 send data bursts to the egress node 5 through the core node 4, respectively. Data bursts from the three ingress nodes should contend for the same resource at the output port of the core node 4. If we assume that data bursts are arriving at a bottleneck node with Poisson distribution and the number of channel is k and traffic loads for node 1, 2 and 3 are ρ_1, ρ_2, ρ_3, respectively, then the burst blocking probability with no buffer can be calculated by using the well-known Erlang loss formula $P_B(k,\ \rho_1+\rho_2+\rho_3)$. But, in the mesh-type networks, the blocking rate will be changed because of the edge/core node combined functions.

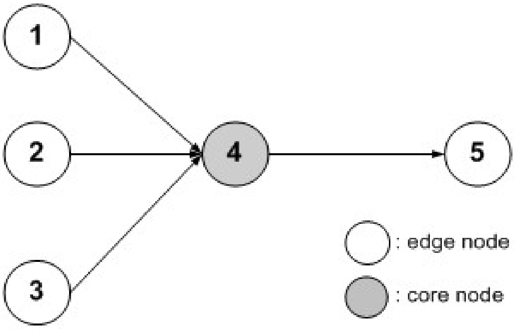


Fig. 1. Network model in the conventional OBS network

2.2 Network Modeling in ECNC OBS Network

In this section, we investigate the performance of the ECNC OBS network as depicted in Figure 2. In Figure 2 (a), the edge/core node combined, the node 4, performs the egress function as well as the core function.

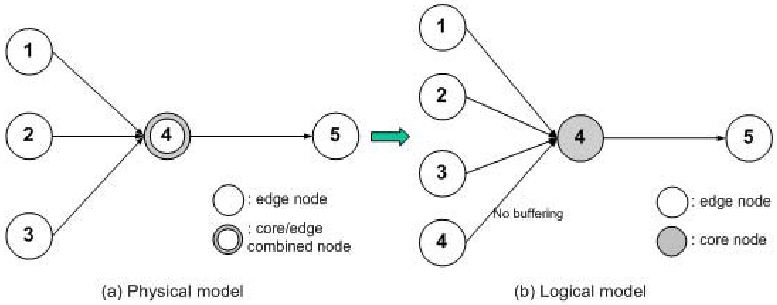


Fig. 2. Network model in the ECNC OBS networks

Thus, it both generates new bursts as an ingress node and cut-though bursts from ingress nodes as the core node. Depicted as the logical model in Figure 2 (b), data bursts from 4 also contend for the outgoing port. Thus, burst blocking probability can be calculated by $P_B(k, \rho_1 + \rho_2 + \rho_3 + \rho_4)$. It is noted that the node 4 do not have buffers. Instead, self-generated data bursts are immediately sent to the outgoing port and contend with TDB after assembled by using the conventional burst assembly schemes [5].

3 Proposed Channel Scheduling Algorithm

3.1 New Scheduling Algorithm for ECNC OBS Network

In the previous chapter, we know that node 4 has the capability to buffer self-generated data bursts for the purpose of void filling between TDB. We propose a new scheduling algorithm for ECNC OBS network to improving throughput as well as reducing the data burst loss rate. In Figure 3, data bursts from 3 ingress nodes contend for the outgoing port of the node 4, however data bursts generated from the node 4 do not contend with TDB

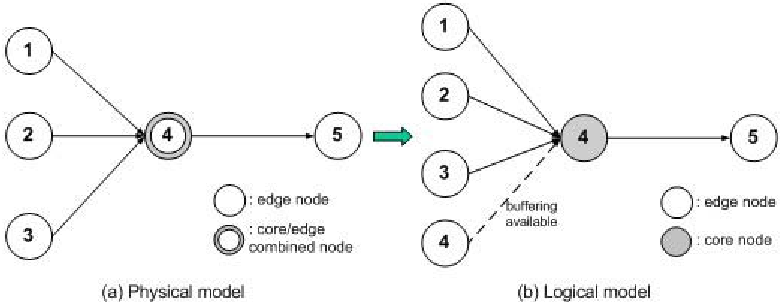


Fig. 3. Network model for the new data scheduling algorithm in edge/core node combined OBS network

but fill void intervals. This void-filling is based on the two capabilities of the node 4, one is the monitoring capability for all voids in the data channel scheduling table and the other is buffering capability to shift data bursts generated by node 4. Thus, the burst blocking probability for all inputs $(\rho_1, \rho_2, \rho_3, and \rho_4)$ can be calculated by $P_B(k, \rho_1 + \rho_2 + \rho_3)$ where ρ_4 has no impact on the loss probability. Compared to loss probability with conventional schemes, $P_B(k, \rho_1 + \rho_2 + \rho_3 + \rho_4)$, the proposed scheduling scheme achieves drastic loss rate reduction for data bursts.

3.2 Throughput of Edge/Core Node Combined Network

To increase throughput, the node uses the void intervals between TDB to transmit one-hop-going (OHG) data bursts. The average burst size of OHG data bursts can be driven by using the analytical method for the proposed network topology. The mean available size of the data burst is the function of the offset-time and the traffic load of the TDB as shown in equation (1) when the offset time of TDB is constant [6].

$$\overline{h_{in}} = f(t_{offset}, \rho_c^{each}) = \int_0^{t_{offset}} x \cdot p(x)dx + \int_{t_{offset}}^{\infty} t_{offset} \cdot p(x)dx \qquad (1)$$

Where, t_{offset} is the offset time of TDB, ρ_c^{each} is traffic load of TDB per each channel, p(x) is the probability that void interval is x, x is the length of void interval, respectively. By using this size of self-generated DB, the throughput of the channel can be calculated analytically with the function of offset-time and traffic load of TDB.

4 Numerical Results

Performance is analyzed based on Figure 2 and Figure 3 for the conventional scheduling algorithm and our proposed data channel scheduling algorithm. Three ingress node and one core node generate data bursts equivalently with Poisson distribution. We assume the data burst size from ingress nodes is same for simplicity. All links consist of 8 wavelengths.

In Figure 4, the burst blocking rate comparisons between the conventional and our proposed data channel scheduling algorithm are presented for the edge/core combined OBS network. It is clearly shown that proposed algorithm has better performance than conventional algorithm because self-generated data bursts do not affect the overall burst blocking rate. Therefore, it is possible to save overall resources to be provided for guaranteeing a certain level of the blocking probability.

To investigate the relationship between offset time TDB and throughput, we compare the throughput of core node by changing the offset time. In Figure 5, while the throughput improvements do not appear in conventional method, the throughput of proposed scheme increases when the offset time ratio to the mean length of TDB changes from 0.2 to 1.0. Throughputs of Figure 5 are acquired by using the equation (1) and the termination rate of void interval of TDB [6]. It means that the throughput will be improved if the offset time of TDB increases for the proposed scheme at the same traffic load.

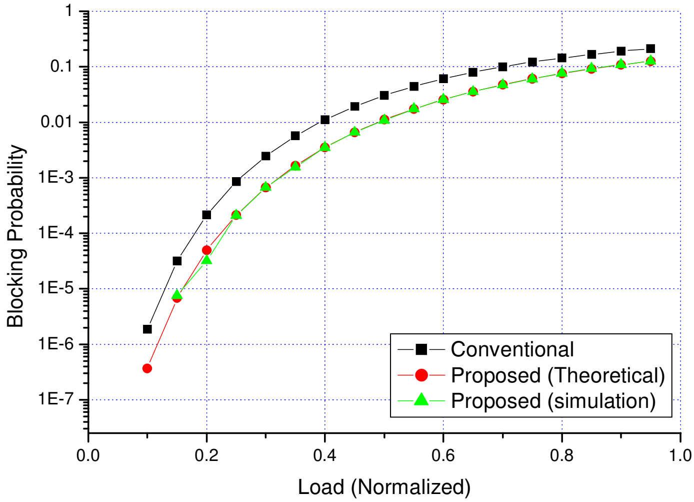


Fig. 4. Data burst blocking rate in conventional and proposed data burst scheduling algorithm in the edge/core combined OBS network

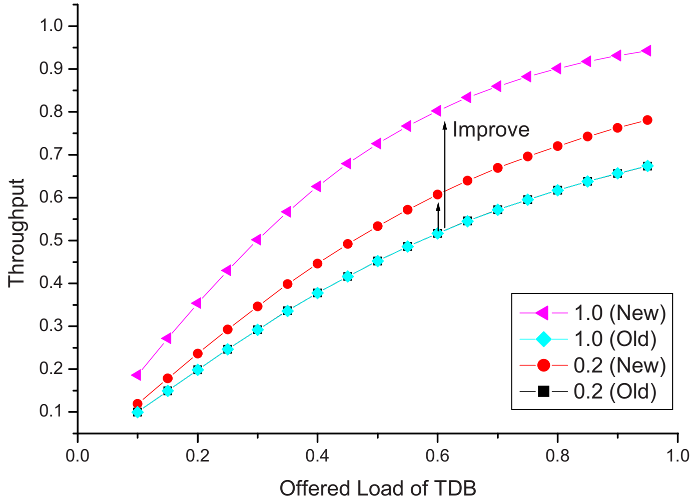


Fig. 5. Relationships between offset-time and throughput

5 Conclusion

In this paper, we propose a new data channel scheduling algorithm for the core/edge node combined OBS network to utilize buffering effect in the ECNC node. If we consider that the nodes perform the edge and core node function, the conventional data channel scheduling algorithm should be modified. Without new scheduling scheme, we cannot achieve any merit of the ECNC OBS nodes. In our proposed data channel scheduling algorithm, the self-generated data bursts at edge/core combined node do not contend with the TDB by using the void intervals and electrical buffering function. Through performance analysis results, it is clear that our proposed scheduling algorithm reduces data burst loss probability and acquires high channel utilization and great benefit to the performance of mesh-type networks.

Acknowledgements

This work was supported in part by the Institute of Information Technology Assessment (IITA) through the Ministry of Information and Communication (MIC) and the Korea Science and Engineering Foundation (KOSEF) through the Ministry of Science and Technology (MOST), Korea.

References

1. M. Yoo and C. Qiao.: A new Optical Burst Switching Protocol for Supporting Quality of Service. *Proc. SPIE All Optical Comm. Syst.: Architecture, Control Network Issues*, Vol. 3531, 1998, pp.395-405.
2. Qiao, C.: Labled Optical Burst Switching for IP-over-WDM Integration. IEEE Communication Magazine, Vol.1, No. 9, September 2000, pp. 104-114.
3. Fei Xue, S.J.B. Yoo, H. Yokoyama and Y. Horiuchi.: Performance comparison of optical burst and circuit switched networks. Optical Fiber Communication Conference, Vol 3, March 2005.
4. X. Huang, V.M. Vokkarane and J.P. Jue.: Burst cloning: a proactive scheme to reduce data loss in optical burst-switched networks. International Conference on Communications, Vol 3, May 2005, pp. 1673 – 1677
5. Y. Xiong, M. Vandenhoute, H. Cankaya.:Control architecture in optical burst-switched WDM networks. *IEEE JSAC,* Vol. 18, 2003, pp. 1838-1851.
6. S.Y. Lee, I.Y. Hwang and H.S. Park.:A New Burst Generation and Scheduling Scheme in Optical Burst Switching Networks. submitted to IEICE

Distributed Real-Time Monitoring with Accuracy Objectives

Alberto Gonzalez Prieto and Rolf Stadler

School of Electrical Engineering,
KTH Royal Institute of Technology, Sweden
{gonzalez, stadler}@ee.kth.se

Abstract. We introduce A-GAP, a protocol for continuous monitoring of network state variables with configurable accuracy. Network state variables are computed from device counters using aggregation functions, such as SUM, AVERAGE and MAX. In A-GAP, the accuracy is expressed in terms of the average error and is controlled by dynamically configuring filters in the management nodes. The protocol follows the push approach to monitoring and uses the concept of incremental aggregation on a self-stabilizing spanning tree. A-GAP is decentralized and asynchronous to achieve robustness and scalability. We provide some results from evaluating the protocol for an ISP topology (Abovenet) in several scenarios through simulation. The results show that we can effectively control the fundamental trade-off between accuracy and overhead. The protocol overhead can be reduced significantly by allowing only small error objectives.

1 Introduction

The ability to provide continuous estimates of management variables is vital for management tasks, such as network supervision, quality assurance, and proactive fault management. Generally, management variables that are monitored in these tasks are aggregates that are computed from device variables across the network using functions such as SUM, AVERAGE, MIN, MAX. Sample aggregates are the total number of VoIP flows in a network domain and the maximum link utilization.

For many management tasks, it is crucial to know how accurate such estimates are. However, network management solutions deployed today usually provide qualitative control of the accuracy, but do not support the setting of an accuracy objective.

Engineering continuous monitoring solutions for network management involves addressing the fundamental trade-off between accurate estimation of a variable and the management overhead in terms of traffic and processing load. Obviously, a high accuracy comes at the cost of a high overhead and, similarly, low accuracy estimation can be achieved with a low overhead. We found this trade-off first discussed in [2]. Since then, several authors addressed this issue as we show in [1].

In this paper, we address the problem of continuous monitoring with accuracy objectives in large-scale network environments. Specifically, we want to achieve an efficient solution that allows us to control the accuracy of the estimation.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1246 – 1251, 2006.

The paper introduces A-GAP, a generic aggregation protocol with controllable accuracy. A-GAP is based on GAP (Generic Aggregation Protocol), which allows for continuously computing aggregates of local variables by (i) creating and maintaining a self-stabilizing spanning tree and (ii) incrementally aggregating the variables [1] (fig 1). A-GAP is push-based in the sense that changes in monitored variables are sent towards the management station. The protocol controls the management overhead by filtering updates that are sent from monitoring nodes to the management station. The filters periodically adapt to the dynamics of the monitored variables and the network environment. All operations in A-GAP, including computing the aggregation function and filter configuration, are executed in a decentralized and asynchronous fashion to ensure robustness and achieve scalability.

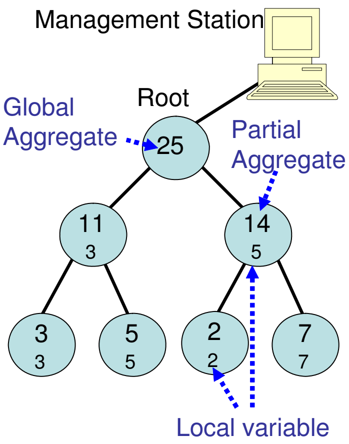


Fig. 1. Example of aggregation tree. Distributed computation of the sum of local variables (w_i).

We developed a stochastic model of the monitoring process, which allows us to compute the filter widths as the solution for the optimization problem of minimizing the management overhead for a given estimation error. A heuristic solution to this problem is implemented in A-GAP.

The paper is organized as follows. Section 2 defines the problem of real-time monitoring with accuracy objectives. Section 3 describes our proposal, A-GAP, which is evaluated in section 4. Section 5 concludes the paper.

2 The Problem: Real-Time Monitoring with Accuracy

System architecture. This work assumes a distributed management architecture, whereby each network device participates in the computation by running a management process, either internally or on an external, associated device. These management processes communicate via a *management overlay network* for the purpose of monitoring. We also refer to this overlay as the *network graph*. A node in this graph represents a management process together with its associated network device(s). While the topology of this overlay can be chosen independently from the topology of the underlying physical network, we assume in this paper, for simplicity, that both topologies are the same, i.e., that the management overlay has the same topology as the as the underlying physical network.

Problem statement. We consider a dynamically changing network graph $G(t) = (V(t), E(t))$ in which nodes $n \in V(t)$ and edges/links $e \in E(t) \subseteq V(t) \times V(t)$ may appear and disappear over time. Each node n has an associated *local variable* $w_n(t) \geq 0$.

The objective is to engineer a protocol on this network graph that provides a management station with a continuous estimate of $\Sigma_n w_n(t)$ for a given accuracy. The accuracy is expressed as the *average error* of the estimate over time. The protocol must minimize the (maximum) processing load across all nodes.

Throughout the paper we use SUM as aggregation function. Other functions can be supported as well, as discussed in section 5.

3 A-GAP: A Distributed Solution

A-GAP controls the management overhead and estimation accuracy by modifying filters in the management nodes. A filter reports changes of the local partial aggregate if its new value exceeds the local filter width. The filter widths periodically adapt to the dynamics of the monitored variables and the network environment. The accuracy objective can be dynamically changed from the management station if needed.

We developed a stochastic model of the monitoring process, which includes the dynamics of the local variables, the filter widths, the overhead incurred and the estimation accuracy. Using this model, we express the filter widths as the decision variables for the problem of *minimizing the maximum load across all nodes in the management overlay for a given average error of the estimation of the global aggregate*. A heuristic solution to this optimisation problem is implemented in A-GAP. The model and the heuristics are described in [1].

In the above model, local variables change following independent random walks. This assumption has been made in similar contexts [1], and it facilitates an algorithmic solution. In practice, the parameters of the random walk must be estimated.

Design principles of A-GAP. First, for reasons of scalability and robustness, A-GAP is a decentralized and asynchronous protocol. Although a centralized solution to the above optimisation problem could be achieved using grid search algorithms, such an approach is not feasible, since its computational complexity grows exponentially with the number of nodes.

Second, A-GAP is an extension of the GAP protocol, which provides continuous estimation of global aggregates by creating a spanning tree on the management overlay and incrementally aggregating the local variables on this spanning tree. When running A-GAP, all nodes of the management overlay execute the same code. The root node of the spanning tree holds the current estimate of the global aggregate.

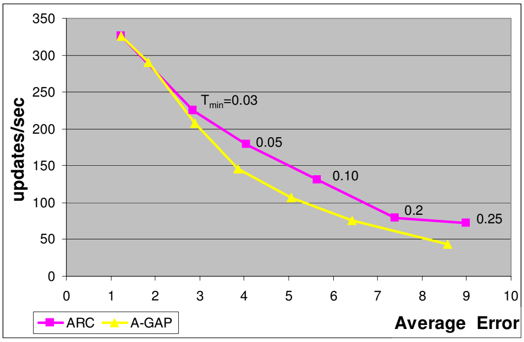


Fig. 2. Management Overhead vs Accuracy for the A-GAP and ARC protocols

Third, A-GAP realizes a heuristic in which the above global problem is mapped onto a local problem that each node solves. Each node attempts to minimize its processing load for a given accuracy regarding its local aggregate. This is achieved by periodically re-computing local filters based on local information.

Re-computing local filters. Each node periodically executes a control cycle in an asynchronous fashion, as follows. The node starts by polling its children for statistics related to their partial aggregates. Then, the node re-computes the filters of a subset of its children. The subset is chosen using a round-robin policy, whereby the sets of two consecutive rounds overlap. The new filters are determined by minimizing the local processing load subject to an accuracy objective for the local partial aggregate. This accuracy objective is given by the node's parent. The problem is solved through a grid search, where the search space is limited to small changes of the current filter width. Next, the new accuracy objectives for the children are computed. Finally, the node updates the statistics of its partial aggregate, which will be polled by its parent during the next control cycle.

4 Evaluation Through Simulation

Setup description. We have evaluated A-GAP through extensive simulations using the SIMPSON simulator [4]. The results presented in the paper are based on the topology of Abovenet [5], consisting of 654 nodes and 1332 links. The overlay topology is chosen to follow the physical Abovenet topology. The control cycle of A-GAP is 1 second. The results reported below corresponds to a measurement period of 30 seconds simulation time of the protocol in steady state, which is reached after a warm-up period of approximately 25 seconds.

Accuracy vs overhead trade-off. In this simulation scenario, the global aggregate increases at an average rate of about 60 units per second. Figure 2 shows the overhead, i.e., the maximum processing load, as a function of the average error. As can be seen, the overhead decreases monotonically, as the error objective is increased. For small errors, the load decreases faster than for larger errors. As expected, the overhead can be reduced by allowing a larger average estimation error. For example, allowing an error of 3 units reduces the load by 40%, an error of 8 units reduces the load by 85%. (Note that the granularity of a local variable is 1 unit).

Figure 2 also compares the performance of A-GAP against an asynchronous rate-control scheme (ARC). As A-GAP, ARC uses a spanning tree and incremental aggregation to continuously estimate the aggregate of local variables. The control parameter for ARC is the minimum time interval (T_{min}) for a node to send an update to its parent. In these and other experiments we performed [1], A-GAP incurred a lower overhead than ARC. As expected, both approaches perform similarly for very small estimation errors.

A rate-control approach that would permit to set update intervals individually for each node would probably perform better than ARC. Note though that, while A-GAP allows to quantitatively control the error objective, rate-control approaches do not support this functionality in a straightforward way.

From this simulation data, we also evaluated the difference between the accuracy objective and the measured error. For all measurement points, we found that the measured error is about 1.5 units above the error objective. We explain this by the fact that updates from different nodes in the network experience different delays for reaching the root of the tree. This difference depends on the network topology and delays.

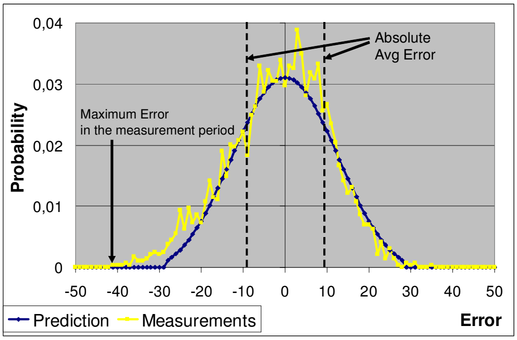


Fig. 3. Distribution of Predicted Errors and Measured Errors at the Root Node.

Distribution of the estimation error. In this simulation scenario, the global aggregate oscillates around a constant value. Figure 3 includes a curve that shows the predicted error based on our stochastic model [1]. The second curve gives the measured error from a simulation run. (The curves correspond to an error objective of 8). A vertical bar indicates the average error.

As we can see, the predicted error distribution is close to the actual distribution. More importantly, the distributions have long tails. While the average error is 9.5, the maximum error in this measurement period is 41 and the maximum possible error (that can occur in an infinite measurement period) is 180. Based on this observation, we argue that an average error objective is more significant for practical scenarios than a maximum error objective, as suggested by other authors (see [1]).

5 Discussion

In this paper, we introduce A-GAP, a protocol for continuous monitoring with accuracy objectives. A-GAP follows the push approach to monitoring and uses the concept of incremental aggregation on a spanning tree. A-GAP is decentralized and asynchronous, two key properties for achieving robustness and scalability.

Although we have used SUM as the aggregation function throughout this paper, other aggregate functions like AVERAGE, MIN and MAX can be supported with straightforward modifications. For instance, AVERAGE can be estimated by maintaining the SUM of the local variables and a node count (obtained using another SUM) at the root.

Our experiments show that we can effectively control the trade-off between accuracy and overhead. A-GAP can reduce the overhead significantly when allowed some error in its estimations.

In A-GAP, accuracy is expressed in terms of the average error, which we argued to be more significant for practical applications than the objective of a maximum error, suggested in the recent literature (see [1]).

To be applicable in practical scenarios, A-GAP requires extensions. Since the model upon which filter re-computation is based does not consider networking and processing delays, the protocol generally misses the error objective by a small margin. In fact, the estimation error of A-GAP exceeds the error objective by a margin that is topology dependent. This means that the protocol needs to be tuned during initialization of the monitoring task.

The model for local filter re-computation uses parameters from the random walk model of the local variables as input. Therefore, these parameters need to be dynamically estimated for each local variable and real-time estimators need to be added to the protocol.

A-GAP enables performance prediction at run-time. Based on our stochastic model, a manager can be provided with an estimation of the expected load on all nodes and the distribution of the estimation error at the root node for a given accuracy objective. This is potentially significant in real scenarios. For instance, a manager could avoid overloading the management system by loosening the error objective.

An implementation of A-GAP with the above mentioned extensions is under way in our laboratory at KTH.

Acknowledgments. The authors would like to thank Mads Dam at KTH for fruitful discussions around the design and evaluation of A-GAP.

This paper describes work undertaken in the context of the Ambient Networks – IST project, which is partially funded by the Commission of the European Union. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the Ambient Networks Project.

References

[1] A. Gonzalez Prieto and R. Stadler, “Distributed Real-time Monitoring with Accuracy Objectives”, KTH Technical Report, December 2005

[2] C. Olston, B. T. Loo and J. Widom, “Adaptive Precision Setting for Cached Approximate Values”, ACM SIGMOD 2001, Santa Barbara, USA, May 2001.

[3] M. Dam, R. Stadler, “A Generic Protocol for Network State Aggregation”, Radiovetenskap och Kommunication (RVK), Linkoping, Sweden, 14-16 June, 2005.

[4] K. Lim and R. Stadler. SIMPSON — a SIMple Pattern Simulator fOr Networks. http://www.comet.columbia.edu/adm/software.htm, 2005.

[5] N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP topologies with Rocketfuel”, ACM/SIGCOMM, 2002, Pittsburgh, USA, August 2002.

Improving Load Balance of Ethernet Carrier Networks Using IEEE 802.1S MSTP with Multiple Regions

Amaro de Sousa and Gil Soares

Institute of Telecommunications, Department of Electronics and Telecommunications,
University of Aveiro, 3810-193 Aveiro, Portugal
asou@det.ua.pt, gsoares@av.it.pt

Abstract. With IEEE 802.1S Multiple Spanning Tree Protocol, an Ethernet operator can define different network regions. A Common Spanning Tree (CST) is defined in such a way that a link failure inside a region does not affect the CST outside the region and additional Spanning Trees inside each region can be configured to achieve better load balance. We propose a procedure to determine the MSTP parameters configuration with multiple regions that optimize load balance and show its efficiency through computational results. We compare multiple region solutions with single region solutions, using a previous work on the single region case, and show that the multiple regions approach is better when traffic is mainly between switches belonging to the same region.

1 Introduction

The IEEE 802.1D STP [1] and IEEE 802.1Q VLAN Protocol [2] are two well established protocols for Ethernet networks. STP routes demands based on a set of active links spanning all switches without cycles, i.e., a Spanning Tree. It includes detection of network changes and Spanning Tree recalculation to recover full connectivity. 802.1Q enables the assignment of traffic demands of different clients to different VLANs in order to prevent packets from one client to reach other client ports. Recently, two new protocols were proposed to enhance the survivability and traffic engineering capabilities of IEEE 802 switching networks. One is the IEEE 802.1W Rapid Spanning Tree Protocol [3], an evolution of STP where port states and roles are redefined and a negotiation mechanism is used to accelerate the convergence to a new Spanning Tree when the network changes. The other is the IEEE 802.1S Multiple Spanning Tree Protocol (MSTP) [4]. With MSTP, a network operator can define different network regions. A Common Spanning Tree (CST) connecting all switches of all regions is set-up in such a way that a link failure inside a region does not affect the CST outside the region. It enables also additional Multiple Spanning Trees to be configured inside each region although limited to support only internal VLANs (VLANs whose ports are in the same region). MSTP does not state how regions should be defined, which Spanning Trees should be created and how VLANs should be assigned to Spanning Trees. There is a trade-off between considering a single region or adopting multiple regions. In terms of failure recovery, the multiple regions case is superior. However, the additional Spanning Trees can only support

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1252 – 1260, 2006.

internal VLANs which limits the load balancing that can be obtained. Previously, we have addressed in [5] the single region case. Other authors [6-8] have addressed the dynamic scenario where VLANs are dynamically requested and Spanning Trees are dynamically set-up and tear-down. In [9], authors address the problem of how to divide the network into regions. Other works propose MSTP as a means to improve the network support of other important aspects like mobility [10] and quality of service [11].

Given (a) a network composed by a set of switches connected through point-to-point links, (b) a set of regions defined on the network and (c) a set of VLANs, each one defined by a set of traffic flows, we determine (i) the appropriate MSTP parameters implementing the CST and the additional Spanning Trees and (ii) the assignment of VLANs to Spanning Trees. The aim is to optimize the network load balance. For any desired Spanning Tree, it is always possible to determine a set of MSTP parameters that makes active its links. Therefore, we propose a procedure that first determines the set of Spanning Trees (together with the mapping of VLANs to Spanning Trees) and, then, determines the MSTP parameters.

2 Solving Procedure

Consider an Ethernet carrier network composed by switches connected through point-to-point links. The network is represented by the directed graph $G = (N,A)$ where N is the set of switches and A is the set of directions (i,j) of all links (E is the set of links $\{i,j\}$). The bandwidth capacity of link $\{i,j\}$ is $b\{i,j\}$. Consider r regions defined on the network, each one identified with a positive number i between 1 and r. Each region is defined by the sub-graph $G_i = (N_i,A_i)$ where $N_i \subset N$ is the set of switches and $A_i \subset A$ the set of direction of links belonging to region i. The network supports a set of VLANs represented by set V. Each VLAN $v \in V$ is characterized by a set of traffic flows $T(v)$ and each traffic flow $t \in T(v)$ is characterized by its origin switch $o(t)$, destination switch $d(t)$ and bandwidth demand $b(t)$.

For a particular set of Spanning Trees and a particular mapping of VLANs to Spanning Trees, each traffic flow is routed through the path defined in the Spanning Tree that its VLAN was assigned to. Assume that $a(v)$ indicates the Spanning Tree instance assigned to VLAN v. Assume a binary parameter $s[(i,j), t]$ that is one if arc (i,j) is in the path of traffic flow $t \in T(v)$ defined by the Spanning Tree instance $a(v)$ assigned to its VLAN v. The load on arc (i,j) is the sum of the demands of all traffic flows that use it. Consider $l(i,j)$ the resulting load (in percentage):

$$l(i,j) = \frac{\sum\limits_{v \in V} \sum\limits_{t \in T(v)} \big(b(t) s[(i,j),t]\big)}{b\{i,j\}} \times 100\% \qquad (1)$$

We define the load array L of a particular solution, the array which is formed by all $l(i,j)$ values sorted in a non increasing order: first element of L is the highest load value; second element is the second highest load value, and so on… In the remaining of this paper, load array L_G is the set of non-increasing link loads on graph G and load array L_{Gi} is the set of non-increasing link loads on sub-graph G_i. Load arrays are used to compare the load balance between different solutions: for two given solutions 1 and 2 whose load arrays are L_1 e L_2, we consider solution 1 better than solution 2 if L_1 has a smaller value than L_2 in the first array position whose values of both arrays are different.

Consider the following additional notation. The set of links forming the CST is given by ϖ where the links inside region i are denoted by ϖ_i and the links outside regions are denoted by ϖ_0 ($\varpi = \varpi_0 \cup \varpi_1 \cup \ldots \cup \varpi_r$). Inside region i, besides the CST ϖ_i, there are n additional Spanning Trees, each one denoted by ϖ_{ij} where $j = 1 \ldots n$. The set of additional Spanning Trees inside region i is denoted by Ω_i ($\Omega_i = \varpi_{i1} \cup \ldots \cup \varpi_{in}$). Array Φ is a VLAN to Spanning Tree assignment array with index $v = 1 \ldots |V|$ where its v^{th} position is $a(v)$, the Spanning Tree assigned to VLAN v. Array Φ is decomposed in r arrays where Φ_i , $i = 1 \ldots r$, refers to the VLANs internal to region i (VLANs not internal to any region are assigned to CST ϖ). In the following description, $\underline{L}_G$ and $\underline{L}_{Gi}$ are the best load arrays obtained respectively in Step 1 and Step 2, $\underline{\varpi}$ is the set of CST links in the best solution, $\underline{\Omega}_i$ is the set of n additional Spanning Trees of region i in the best solution and $\underline{\Phi}$ is the VLAN to Spanning Tree assignment array of the best solution. The proposed procedure is composed of three steps (presented in the next box).

```
1:  Set all values of $\underline{L}_G$ to $+\infty$                                  // Begin of Step 1
2:  while Number of Iterations < MaxMain do:
3:          $\varpi \leftarrow GenerateCST(G)$
4:          $(L_G , \varpi) \leftarrow ImproveCST(\varpi)$
5:          if $L_G$ is better than $\underline{L}_G$ do:
6:              $\underline{L}_G \leftarrow L_G$ , $\underline{\varpi} \leftarrow \varpi$                     // End of Step 1
7:  for $i = 1 \ldots r$ do:                                                  // Begin of Step 2
8:          Set all values of $\underline{L}_{Gi}$ to $+\infty$
9:          while Number of Iterations < MaxRegion do:
10:             $(\varpi_i , \Omega_i) \leftarrow GenerateMST(n+1 , G_i)$
11:             $(L_{Gi} , \Phi_i) \leftarrow AssignVLANtoTrees(\varpi_i , \Omega_i)$
12:             if $L_{Gi}$ is better than $\underline{L}_{Gi}$ do:
13:                 $\underline{L}_{Gi} \leftarrow L_{Gi}$ , $\underline{\Omega}_i \leftarrow \Omega_i$ , $\underline{\Phi}_i \leftarrow \Phi_i$ , $\underline{\varpi}_i \leftarrow \varpi_i$    // End of Step 2
14: DetermineCSTParameters($\underline{\varpi}$ , $G$)                                 // Begin of Step 3
15: for $i = 1 \ldots r$ do:
16:     for $j = 1 \ldots n$ do:
17:         DetermineSTParameters($\underline{\varpi}_{ij}$ , $G_i$)                       // Begin of Step 3
```

In Step 1, we determine the CST set of links optimizing the resulting L_G array. Step 1 procedure runs *MaxMain* iterations (**while** cycle from line 2 to 6). On each iteration, it generates one Spanning Tree over the graph G (procedure *GenerateCST*); it then improves this Spanning Tree with respect to the link load array L_G (procedure *ImproveCST*) and, if the resulting load array L_G is better than the best load array $\underline{L}_G$ (line 5), it saves the present Spanning Tree as the best solution found so far (line 6). In step 2, we solve each region separately; we determine the set of links of CST inside the region and of all additional Spanning Trees optimizing the resulting L_{Gi} array. Step 2 procedure is executed one time for each region (**for** cycle from line 7 to line 13). For each region, Step 2 procedure runs *MaxRegion* iterations (**while** cycle from line 9 to 13). On each iteration, it generates a set of n+1 Spanning Trees over graph G_i (procedure *GenerateMST*); it determines an assignment array Φ_i indicating the

Spanning Tree assigned to each VLAN, together with the resulting load array L_{Gi} (procedure *AssignVLANtoTrees*) and if load array L_{Gi} is better than the best load array $\underline{L}_{Gi}$ (line 12), it saves the present set of Spanning Trees as the best solution found so far (step 13). In Step 3, we determine the appropriate MSTP protocol parameters. The procedure determines the MSTP parameters for the CST set of links over graph G (procedure *DetermineCSTParameters*) and for all additional Spanning Trees (line 16) of all regions (line 15) over the corresponding graph G_i (procedure *DetermineSTParameters*). In the remaining of this section, the elementary procedures defined on each step are separately described.

There are two procedures for the generation of Spanning Trees. The aim of procedure *GenerateCST*(G) in Step 1, is to randomly generate a set of links ϖ that can define a CST. In order to agree with MSTP, this set of links must be an in-region Spanning Tree, i.e., it should form a Spanning Tree on each sub-graph G_i (see example in Fig. 1). We generate a random spanning tree as follows. First, we assign all nodes with a different label. Then, we repeat $|N| - 1$ times the following operations: (i) select randomly one link among all links whose end-nodes have different labels and (ii) assign all nodes with the label of one end-node with the label of the other end-node. In order to guarantee the in-region property, procedure *GenerateCST*(G) first selects the links of each region and, then, proceed to the links outside all regions.

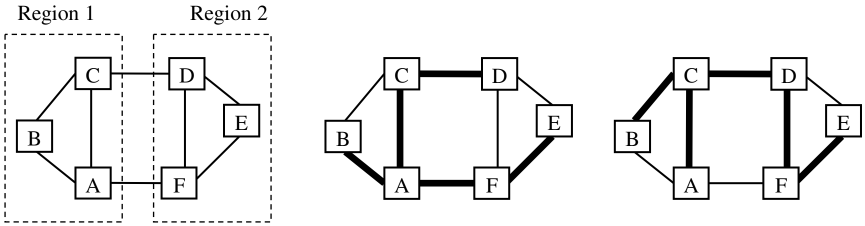


Fig. 1. In the middle, the thick links do not define a proper CST since they do not form a Spanning Tree inside Region 2. In the right, the thick links define a proper CST.

The aim of procedure *GenerateMST*(n+1,G_i) in Step 2, is to generate for the region defined by G_i a set of n+1 Spanning Trees avoiding common links. The Spanning Trees to be generated are ϖ_i (the CST part internal to region i) and the n internal Spanning Trees $\Omega_i = \varpi_{i1} \cup \ldots \cup \varpi_{in}$. The reason for avoiding common links is twofold: it helps splitting the traffic flows among more links, which results in a better load balanced network, and it minimizes the impact of link failures inside the region since a link failure only affects the traffic flows assigned to the Spanning Trees that use the failed link. We avoid common links in a greedy way: we generate randomly the first Spanning Tree; then, we generate randomly the second Spanning Tree using as much as possible the links not used in the first one; etc…

The aim of procedure *ImproveCST*(ϖ) in Step 1 is, based on the current Spanning Tree ϖ, to determine a better Spanning Tree ϖ and to determine its load array L_G. This procedure is based on a local search technique with the following neighbor definition: a Spanning Tree ω' is a neighbor of a given Spanning Tree ω if it differs from ω only in a single link. $V(\omega)$ designates the set of all neighbors of ω with the in-region Spanning Tree property. In the following description, ω and ω' are auxiliary Spanning Trees and $\underline{L}_G$ is the best load array obtained at the end of the procedure.

Procedure *ImproveCST*(ϖ)

```
Determine $\underline{L}_G$ assigning all VLANs to $\varpi$
repeat
    $\omega \leftarrow \varpi$
    for ($\omega' \in V(\omega)$) do:
        Determine $L_G$ assigning all VLANs to $\omega'$
        if ($L_G$ is better than $\underline{L}_G$) do:
            $\underline{L}_G \leftarrow L_G$ , $\varpi \leftarrow \omega'$
until ($\varpi$ equal to $\omega$)
return($L_G$ , $\varpi$)
```

The aim of procedure *AssignVLANtoTrees*(ϖ_i,Ω_i) in Step 2 is to determine a VLAN to Spanning Tree assignment array **Φ** that minimizes the resulting load array L_{Gi}. Note that from the previously determined CST in Step 1, it is possible to determine the flows of external VLANs that cross region *i* and, for each of them, their incoming region node (if their origin node is outside the region) and their outgoing region node (if their destination node is outside the region). Consider V_{int} the set of internal VLANs. This procedure first generates a random assignment array and, then, performs a local search algorithm with the following neighbor structure: an assignment array **Φ'**= $a'(1\ldots|V_{int}|)$ is a neighbor of a given **Φ**= $a(1\ldots|V_{int}|)$ if it differs from **Φ** only in a single element. In the following description, **Φ'** and **Φ''** are two auxiliary VLAN to Spanning Trees assignment arrays.

Procedure *AssignVLANtoTrees*(ϖ_i , Ω_i)

```
for ($v = 1\ldots|V_{int}|$) do:
    $a(v) \leftarrow$ random($0\ldots n$)
Determine $\underline{L}_{Gi}$ assigning external VLANs to 0 and internal VLANs $v$ according to
Φ
repeat
    $\Phi' \leftarrow \Phi$
    for ($v = 1\ldots|V_{int}|$) do:
        for ($p = 0\ldots n$ and $p \neq a(v)$) do:
            $\Phi'' \leftarrow \Phi'$
            $a''(v) \leftarrow p$
            Determine $L_{Gi}$ assigning external VLANs to 0 and internal VLANs
            $v$ according to $\Phi''$
            if ($L_{Gi}$ is better than $\underline{L}_{Gi}$) do:
                $\Phi \leftarrow \Phi''$ , $\underline{L}_{Gi} \leftarrow L_{Gi}$
until ($\Phi$ equal to $\Phi'$)
return($L_{Gi}$ , $\Phi$)
```

There are two procedures for the determination of MSTP parameters. Each $(k,l) \in A_i$ has as associated forwarding port on switch k towards switch l whose *PortCost* must be determined. Given a desired Spanning Tree ϖ_{ij} over graph G_i and a current set of *BridgeID* and *PortCost* values, procedure *DetermineSTParameters*(ϖ_{ij},G_i) updates these values in order to make active the links of ϖ_{ij}:

1. Keep all *BridgeID* values unchanged. The switch with lowest *BridgeID* is the Root Bridge.
2. The forwarding ports in the path defined by ϖ_{ij} from every switch to the Root Bridge are root ports. Keep the *PortCost* values of root ports unchanged.
3. For each switch k, determine its root path cost c_k as the sum of the *PortCost* values of all root ports in the path from k to the Root Bridge.
4. For all non root ports of all links $(k,l) \in A_i$, assign a *PortCost* value equal to $c_k - c_l + 1$ if this value is lower than its current *PortCost* value; if not, let the current *PortCost* value unchanged.

The aim of procedure *DetermineCSTParameters*(ϖ,G) is similar to the previous one but its implementation is more complex. MSTP [4] defines that (i) the switch with lowest *BridgeID* becomes the Root Bridge, (ii) the root path cost of a path from any switch to the Root Bridge (if it is not in the same region as the Root Bridge) considers only the *PortCost* values of the root ports belonging to links outside regions (e.g., the *PortCost* values of links inside regions are considered NULL), (iii) at each region, the Bridge with lower root path cost to the Root Bridge becomes the Regional Root Bridge and (iv) inside each region, the active links are the ones in the minimum cost paths from each switch to its Regional Root Bridge. The following procedure minimizes the number of parameters to be updated:

1. Keep all *BridgeID* values unchanged. The switch with lowest *BridgeID* is the Root Bridge.
2. The forwarding ports in the path defined by ϖ from every switch to the Root Bridge are root ports. Keep the *PortCost* values of root ports unchanged.
3. For each switch k determine its root path cost c_k as the sum of the *PortCost* values of the root ports not internal to any region in the path from k to the Root Bridge.
4. For all non root ports of links (k,l) not internal to any region, assign a *PortCost* value equal to $c_k - c_l + 1$ if this value is lower than its current value; if not, let the current *PortCost* value unchanged.
5. For each region $i = 1 \ldots r$ do:
   - 5.1. The switch belonging to region i with the lowest root path cost is the Regional Root Bridge.
   - 5.2. For each switch k of region i, determine its regional root path cost α_k as the sum of the *PortCost* values of all root ports in the path from k to the Regional Root Bridge.
   - 5.3. For all non root ports of links (k,l) internal to region i, assign a *PortCost* value equal to$\alpha_k - \alpha_l + 1$ if this value is lower than its current *PortCost* value; if not, let the current *PortCost* value unchanged.

3 Computational Results

The proposed procedure was implemented in C and run in a PC, 2.0 GHz Pentium 4 processor, 512 MB of RAM. The case studies consider the network shown in Fig.2. Concerning traffic demands, we have considered 51 VLANs (each VLAN with 2 traffic flows) randomly selected between all access switch pairs and all access-gateway pairs. Demand values were randomly selected among 4 different values (10, 20 50 and 100 Mbps) considering three different case studies: the percentage of total demand internal to regions is 20% for case study A, 50% for case study B and 80% for case study C.

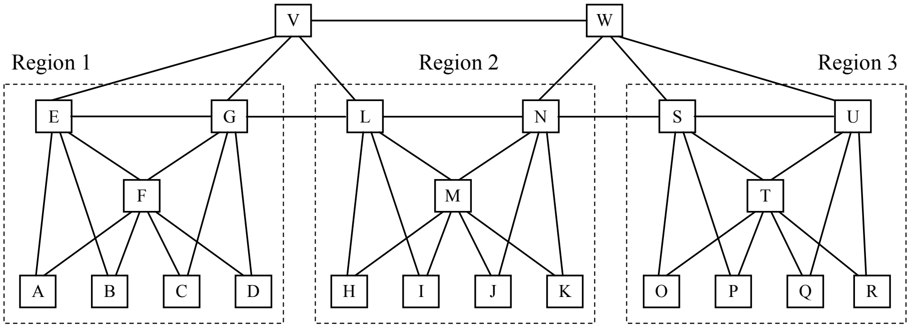


Fig. 2. An Ethernet network with 23 nodes, 42 links and 3 regions (all links have a capacity of 1 Gbps). Switches A to D, H to K and O to R are access switches where customer equipment is connected to and switches V and W are gateway switches that connect the Ethernet network to other core networks.

We have solved all three case studies considering a number *n* of additional Spanning Trees inside each region equal to 1 and 2. In all cases, the *n* = 2 case did not find a solution better than the best solution for the *n* = 1 case. Table 1 shows the obtained results. In case study A, most of the worst load values are on links outside the regions (as we can see in the *n* = 0 line) and since multiple Spanning Trees can only improve the load balance inside regions, there is only a minor improvement in using one additional Spanning Tree. On the other end, the additional Spanning Tree provides a significant improvement in case study C (worst link loads are decreased from 41% to 29%) since all worst load values in the single CST best solution correspond to links inside regions. Case study B is in the middle of the two previous results: the additional Spanning Tree enabled a small decrease of the worst link loads from 55% to 51%.

All cases were solved with both *MaxMain* and *MaxRegion* set to 10000. The procedure Step 1 part took at most 160 seconds with the best solutions always found within the first 10 iterations for all cases (below 0.1 seconds). The procedure Step 2 part took at most 2 seconds with the best solutions always found within the first 1000 iterations for all cases (below 0.2 seconds). These results show that the proposed procedure is very efficient and able to obtain solutions within short computing times.

Table 1. Highest 10 values of load array L_G for each case study; $n = 0$ corresponds to the best solution at the end of Step 1, i.e., the best CST configuration with no addtitional Spanning Trees; $n = 1$ corresponds to the best solution found with 1 additional Spanning Tree on each region. The values marked with * correspond to region internal links.

		Index of first 10 positions of Load Array (%)									
Case Study	*n*	1	2	3	4	5	6	7	8	9	10
A	0	76	76	70*	70*	68	68	58	58	46*	46*
	1	76	76	68	68	59*	59*	58	58	40*	40*
B	0	55*	55*	51	51	49	49	46*	46*	43	43
	1	51	51	49	49	43	43	36*	36*	23*	23*
C	0	41*	41*	39*	39*	34*	34*	32*	32*	32*	32*
	1	29	29	22*	22*	22*	22*	21*	21*	20*	20*

In order to compare both approaches, we have used the algorithm proposed in [5] to obtain the best solutions for the three case studies assuming the whole network as a single region (Table 2). Consider first the case study C. In this case, most of the traffic is internal to regions and the solution for the 3 regions approach shown in Table 1 (with a worst load value of 29%) is only slightly worst than the best solution with the single region approach shown in Table 2 (with a worst load value of 21%). In this case, the 3 region approach is a good solution since it can achieve a load balance almost as good as the optimal single region solution and it minimizes link failure impact on the network. In the A and B case studies, the single region approach obtains significantly better load balanced solutions (worst link values decrease from 76% to 31% in case study A and from 51% to 23% in case study B). Note that in case study A, the 3 regions approach is even worst than the IEEE 802.1D STP solution (worst load is 70%) and this is because the STP case does not require a set of active links with the in-region Spanning Tree property. In these cases, there is a trade-off between load balance and link failure impact.

As a final conclusion, the results shown in Table 2 show that near optimal solutions were obtained with 2 additional Spanning Trees and negligible gains were obtained with 3 additional Spanning Trees. Remember also that in the previous section 1 additional Spanning Tree was enough to obtain the best load balance. These observations show that it is possible to optimize load balance with a small number of Spanning Trees, thus, not penalizing significantly the switches processing overhead.

Table 2. Highest 10 values of load array L_G for each case study considering the whole network as a single region; the $n = 1$ case corresponds to the standard IEEE 802.1D STP protocol

		Index of first 10 positions of Load Array (%)									
Case Study	*n*	1	2	3	4	5	6	7	8	9	10
A	1	70	70	68	68	59	59	58	58	52	52
	2	31	31	31	31	30	30	30	30	30	30
	3	31	31	31	31	30	30	26	26	25	25
B	1	55	55	51	51	46	46	43	43	40	40
	2	23	23	23	23	21	21	21	21	21	21
	3	23	23	23	23	20	20	20	20	20	20
C	1	41	41	39	39	34	34	32	32	32	32
	2	21	21	20	20	20	20	19	19	18	18
	3	21	21	20	20	20	20	19	19	17	17

4 Conclusions

In this paper, we have addressed the problem of how to use the IEEE 802.1S MSTP to improve load balance in Ethernet carrier networks. We have addressed the multiple region case and we have proposed a procedure to determine the MSTP parameters configuration that optimizes network load balancing. The computational results show that this procedure is very efficient and able to obtain solutions within short computing times. We have compared how good the multiple region case is in terms of load balancing. We have showed that (i) the multiple regions approach is the best solution when traffic is mainly between switches belonging to the same region (minimizes link failure impact while achieving good load balance) but (ii) both approaches represent a trade-off between load balance and link failure impact when traffic is more uniformly distributed.

References

1. IEEE 802.1D, “Media Access Control (MAC) Bridges” (1998)
2. IEEE 802.1Q, “Virtual Bridged Local Area Networks” (1998)
3. IEEE 802.1W, “Part 3: Media Access Control (MAC) Bridges – Amendment 2: Rapid Reconfiguration” (2001)
4. IEEE Standard 802.1S, “Virtual Bridged Local Area Networks – Amendment 3: Multiple Spanning Trees (2002)
5. de Sousa, A.: Improving Load Balance and Resilience of Ethernet Carrier Networks with IEEE 802.1S Multiple Spanning Tree Protocol. 5th Int. Conference on Networking (ICN’06), Mauritius Islands (2006)
6. Ali, M., Chiruvolu, G., Ge, A.: Traffic Engineering in Metro Ethernet. IEEE Network Vol. 19 No. 2 (2005) 10–17
7. Kolarov, A., Sengupta, B., Iwata, A.: Design of Multiple Reverse Spanning Trees in Next Generation of Ethernet-VPNs. IEEE GLOBECOM’04, Dallas, USA Vol. 3 (2004) 1390–1395
8. Sharma, S., Gopalan, K., Nanda, S., Chiueh, T.: Viking: A Multi-Spanning-Tree Ethernet Architecture for Metropolitan Area and Cluster Networks. IEEE INFOCOM’04, Hong Kong, Vol. 4 (2004) 2283–2294
9. Padmaraj, M., Nair, S., Marchetti, M., Chiruvolu, G., Ali, M.: Traffic Engineering in Enterprise Ethernet with Multiple Spanning Tree Regions. Proc. of System Communications (ICW’05), Montreal, Canada (2005) 261–266
10. Ishizu, K., Kuroda, M., Kamura, K.: SSTP: an 802.1s Extension to Support Scalable Spanning Tree for Mobile Metropolitan Area Network. IEEE GLOBECOM’04, Dallas, USA Vol. 3 (2004) 1500–1504
11. Lim, Y., Yu, H., Das, S., Lee, S.-S., Gerla, M.: QoS-aware multiple spanning tree mechanism over a bridged LAN environment. IEEE GLOBECOM’03, San Francisco, USA Vol. 6 (2003) 3068–3072

A Simple Sink Mobility Support Algorithm for Routing Protocols in Wireless Sensor Networks

Chun-Su Park, You-Sun Kim, Kwang-Wook Lee,
Seung-Kyun Kim, and Sung-Jea Ko

Department of Electronics Engineering, Korea University,
Anam-Dong Sungbuk-Ku, Seoul, Korea
{cspark, yskim, kwlee, skkim, sjko}@dali.korea.ac.kr

Abstract. In order to support the sink mobility of conventional routing protocols, we propose a simple route maintaining algorithm which does not use the flooding method. In the proposed method, when the sink loses the connection with the source, it does not rebuild an entire route but simply repairs the existing route based on local information. Experimental results show that the proposed algorithm drastically improves the conventional routing protocols in terms of both energy and delay in case of mobile sink.

1 Introduction

Energy is the most crucial resource in the wireless microsensor networks due to the difficulty of recharging batteries of thousands of devices in remote or hostile environments. When a sink is mobile, the energy is consumed for building new packet forwarding route, disseminating data, and maintaining linkage between source and sink. The more frequently the sink moves, the more energy is consumed to maintain the linkage between the source and the sink. Some routing protocols have been recently proposed to support the mobile sink [1], [2], [3], [4]. Instead of flooding query packets, Scalable Energy-efficient Asynchronous Dissemination protocol (SEAD) constructs and maintains a data dissemination tree from source to multiple mobile sink [1]. A sink that wants to join the tree registers itself with the closest access node. When the sink moves out of range of the access node, the route is extended through the inclusion of a new access node. A Two-Tier Data Dissemination (TTDD) was proposed to provide scalable and efficient data delivery to mobile sinks [2]. Upon detecting a stimulus, each source node proactively builds a grid structure which enables a mobile sink to receive data continuously while moving by flooding queries within its local cell only. These protocols have some defects in their own assumptions and network model. For example, each sensor node is assumed to be aware of its own geographic location and mobile sensor nodes are not allowed. Moreover, SEAD and TTDD constrain the method of building the data forwarding route in order to support mobile sink. These constraints make them very difficult to be adopted in the other routing protocols for sink mobility.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1261–1266, 2006.

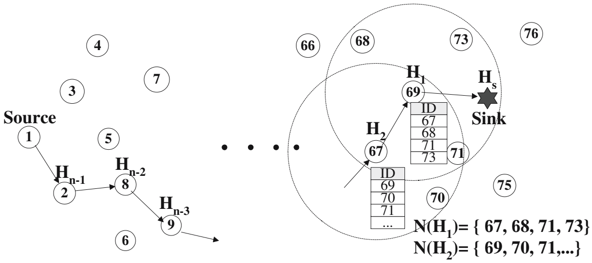


Fig. 1. Examples of H_k and $N(H_k)$

In this paper, in order to support the sink mobility of the conventional routing protocols, we propose a simple sink mobility support (SMS) algorithm which does not use the flooding method. The proposed algorithm can be easily adopted in most existing routing protocols since it does not need to know the geometric location of sensor nodes. Moreover, the proposed algorithm incurs very few communication overhead. The rest of this paper is organized as follows. Section 2 presents network model and the proposed algorithm. Experimental results are presented in Section 3.

2 Network Model and SMS

2.1 Network Model

For the SMS algorithm, we adopt the following assumptions: First, there is multihop data transmission between source and sink. Second, each sensor has a limited battery energy. Third, the speed of the mobile sink is limited. And last, the sensor network has a sufficient number of sensor nodes. For the sake of explanation, we limit our consideration to the case of a single sink. The proposed method can be easily extended to the multiple sink. The target microsensor network model is represented by a set U consisting of scattered sensor nodes. Let $R = \{H_1, H_2, ..., H_k, ..., H_{n-1}, H_n\} \subset U$ be the set of the sensor nodes existing along the data forwarding path from source to sink, where k represents the node distance from the sink on a hop scale. For example, in Fig. 1, $k = 1$ at *Node 69* and $k = 2$ at *Node 67*. Two nodes are said to be *neighbor* if they can directly communicate with each other within a single hop. Whenever a node first receives any packet from its neighboring node, it registers the ID of the neighboring node in its own *neighbor table*. However, if a node does not receives any response from a certain neighboring node during a fixed time, it removes the ID of the neighboring node from its own *neighbor table*. Let $N(H_k)$ be the set of nodes in the *neighbor table* of H_k. Fig. 1 shows $N(H_1) = \{67, 68, 71, 73\}$ and $N(H_2) = \{69, 70, 71, ...\}$. Each node in R generates the description table of a current sensing task in which it participate [5], [6]. In most routing protocols, the sensor node relays received data packets to its own downstream node using the *task description table* that contains source ID, sink ID, data type, its own downstream node ID, and so on. If there are more than one sink and source, each data forwarding path can be distinguished using the *task description table*.

2.2 The SMS

The SMS consists of three phases; preliminary investigation, node selection, and route correction.

Preliminary Investigation. Before a sink moves out of the radio range of H_1, it must store neighbor tables of some H_k's close to the sink. Let $\Omega = \{N(H_1), N(H_2), ...\}$ be a set of neighbor tables to be stored in the sink. The Ω is used for future route correction. Note that the neighbor tables of H_1 and H_2 are sufficient to support mobile sinks such as human being, robots, and tanks. Thus, we focus on the case of $\Omega = \{N(H_1), N(H_2)\}$. For example, in Fig. 1, the sink gathers the neighbor tables from *Node 69* and *Node 67*.

Node Selection. Next, we describe how to maintain connectivity between source and mobile sink. When the sink finds out that it loses the connection with H_1 due to its movement, it broadcasts the *Get_Near_Node* message containing the task description table to search sensor nodes nearby. All nodes that received the *Get_Near_Node* message send a response message to the sink. Then, the sink generates or updates the neighbor table $N(\widetilde{H}_s)$ consisting of the ID's of the nodes which have transmitted response messages. Among $N(\widetilde{H}_s)$, the sink chooses a new H_1, $\widetilde{H}_1$, that has the smallest hop distance to the source by using $N(\widetilde{H}_s)$ and Ω. Let the number of elements in set G be $n[G]$. Then, $\widetilde{H}_1$ can be selected using the following procedure:

(a) If $n[R \cap N(\widetilde{H}_s)] = 1$, then $R \cap N(\widetilde{H}_s) = \{H_{k_1}\}$ and H_{k_1} becomes $\widetilde{H}_1$ for the sink. If $n[R \cap N(\widetilde{H}_s)] = m$, with $m \geq 2$, $R \cap N(\widetilde{H}_s) = \{H_{k_1}, H_{k_2}, ..., H_{k_m}\}$. Then, among H_{k_i}'s, the one that has the smallest hop distance to the source becomes $\widetilde{H}_1$. For example, if $\{H_2, H_3\} \subset R \cap N(\widetilde{H}_s)$, the sink selects H_3 as $\widetilde{H}_1$. If $R \cap N(\widetilde{H}_s) = \phi$, go to (b).

(b) If $n[N(H_2) \cap N(\widetilde{H}_s)] = 1$, then $N(H_2) \cap N(\widetilde{H}_s) = \{H_{k_1}\}$ and H_{k_1} becomes $\widetilde{H}_1$ for the sink. If $n[N(H_2) \cap N(\widetilde{H}_s)] = m$, with $m \geq 2$, then $N(H_2) \cap N(\widetilde{H}_s) = \{H_{k_1}, H_{k_2}, ..., H_{k_m}\}$. In the case, the nodes receiving *Get_Near_Node* message send a response messages containing the information on their own remaining energy to the sink. Then, among H_{k_i}'s, the one that has the largest energy becomes $\widetilde{H}_1$. If $N(H_2) \bigcap N(\widetilde{H}_s) = \phi$, go to (c).

(c) In the same manner as (b), the sink selects $\widetilde{H}_1$ among $N(H_1) \bigcap N(\widetilde{H}_s)$. And if $N(H_1) \bigcap N(\widetilde{H}_s) = \phi$, go to (d).

(d) In this case, there is no candidate for $\widetilde{H}_1$, i.e., $R \cap N(\widetilde{H}_s) = N(H_2) \cap N(\widetilde{H}_s) = N(H_1) \bigcap N(\widetilde{H}_s) = \phi$. Thus, the sink must set up a new route toward the source. The rebuilding method is the same algorithm that the original routing protocol adopted in the wireless sensor network follows. This case happens when the sink enters the empty regions, moves too fast, or a large number

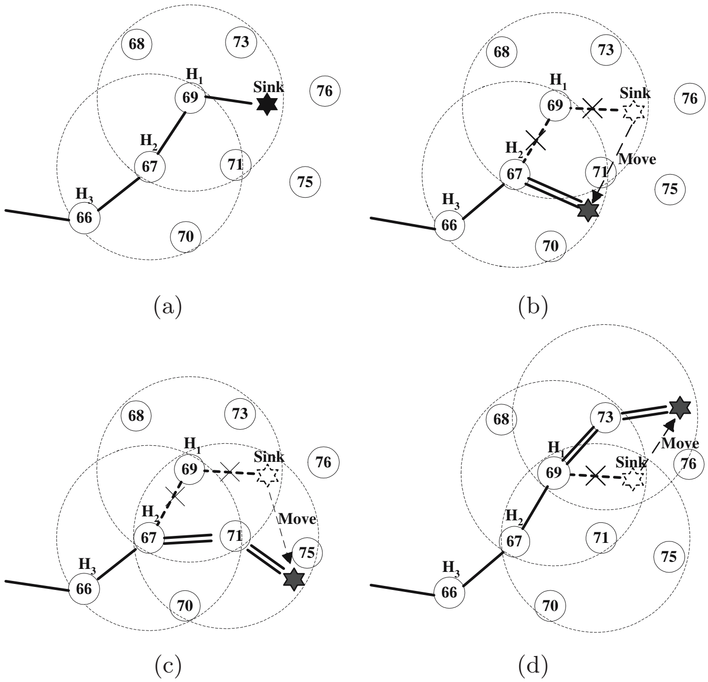


Fig. 2. Types of route correction. In these figures, solid, dot, and double solid lines represent the existing, broken, and newly created paths, respectively. (a) Initial microsensor network. (b) Type 1. (c) Type 2. (d) Type 3.

of nodes run out of their own batteries. But, in general, this case seldom happens and can be reduced or eliminated by expanding the range of *neighbor table* from one hop to two hops or more.

Route Correction. The last phase of the SMS corrects the broken route caused by the sink movement. There are 3 types of route correction.

Type 1: If $\widetilde{H}_1 \in R$, the sink transmits the *Route_Update* message to $\widetilde{H}_1$. Then, $\widetilde{H}_1$ updates its data forwarding path from its downstream node to the sink (see Fig. 2(b)).

Type 2: If $\widetilde{H}_1 \in N(H_2)$, the sink must transmit to $\widetilde{H}_1$ the *Route_Update* message including the address of $\widetilde{H}_2$ since $\widetilde{H}_1$ does not know the address of its upstream node $\widetilde{H}_2$ from which $\widetilde{H}_1$ will receive data packets. $\widetilde{H}_1$ relays the *Route_Update* message to $\widetilde{H}_2$. Then $\widetilde{H}_1$ and $\widetilde{H}_2$, respectively, correct their own data forwarding paths to the sink and $\widetilde{H}_1$ (see Fig. 2(c)).

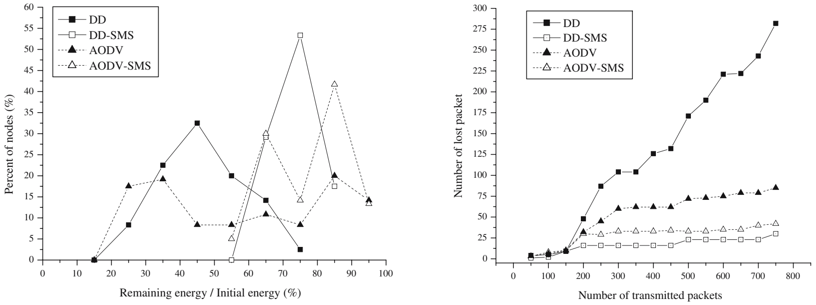
Type 3: If $\widetilde{H}_1 \in N(H_1)$, $\widetilde{H}_1$ relays the *Route_Update* message from the sink to $\widetilde{H}_2$ in the same manner as Type 2 (see Fig. 2(d)).

In route correction phase, the sink sends the *Route_Update* message to the $\widetilde{H}_1$ in order to correct the old route. If $\widetilde{H}_1$ and $\widetilde{H}_2$ send to the sink the response messages including their own neighbor tables, the future preliminary investigation phase for the next movement of the sink can be conducted in current route correction phase, simultaneously.

3 Experimental Result

The performance of the SMS is evaluated using the NS-2 simulator [7]. Our simulation uses the power consumption model that requires 0.660W for transmitting and 0.395W for receiving and 0.035W for idle. The target wireless microsensor network consists of 120 sensor nodes in a 1000m × 1000m field. The transceiver has a 150m radio range and the energy consumption is measured in terms of Joules/node. In our experiment, a single mobile sink is moving at 10 m/sec, i.e., the fastest human speed and the simulation time is 1000 sec. Since the repaired route may not be a globally optimized one, the sink does not perform route correction but rebuilds the entire route whenever the sink has moved thirty times.

In our experiment, we combined the most famous routing protocols, Ad Hoc On Demand Distance Vector (AODV) and Direct Diffusion (DD), with the proposed SMS. These two upgraded versions are named DD-SMS and AODV-SMS. Fig. 3(a) is a graph showing the distribution of the remaining energy for each protocol. In DD, the period for interest packets is set to 5 sec. Since the DD performs flooding to make a new routing table, the remaining energy of all nodes is small and its variance is relatively even over the whole network. As shown in Fig. 3(a), the remaining energy of DD is distributed within a band between 15% ∼ 75%. In case of AODV, the sink broadcasts query packets throughout the network to rebuild an entire route. In addition, unlike DD, AODV uses *Hello packet* to search neighboring nodes, which causes additional energy consumption. Thus, AODV is less efficient than DD in respect of energy consumption. In Fig. 3(a), the measured remaining energy of AODV is between 15% ∼ 95%. Since DD-SMS and AODV-SMS do not use the flooding method and route rebuilding is performed only inside the limited local region, they are more energy efficient



(a) (b)

Fig. 3. (a) Remaining energy. (b) Packet delivery.

than the original DD and AODV protocols. In our experiment, the measured remaining energy of DD-SMS is distributed within a narrow band between 55% ∼ 85%, and that of AODV-SMS is distributed between 55% ∼ 95%. The average remaining energy of DD-SMS and AODV-SMS is improved about 40% as compared with that of DD and AODV.

We also compare the number of lost packets of DD and AODV with that of DD-SMS and AODV-SMS. Fig. 3(b) shows the number of lost packets versus the number of transmitted packets. DD performs worst among the above mentioned four routing protocols in all cases. This is because a new route is made after the sink floods interest packets at the fixed time, i.e., the disconnection time of DD is longer than that of any other protocol. Fig. 3(b) shows that the combination of DD with SMS (DD-SMS) can significantly improve the performance of the DD. By adopting proposed SMS, the packet loss ratio of the AODV and DD is decreased by about 47% and 85%, respectively.

References

1. H. S. Kim, T. Abdelzaher, and W. H. Kwon,: Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks, ACM Conference on Embedded Networked Sensor Systems (2003) 193–204
2. Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, and Lixia Zhang,: TTDD: Two-tier Data Dissemination in Large-scale Wireless Sensor Networks, ACM/Kluwer Mobile Networks and Applications (MONET), Special Issue on ACM MOBICOM (2003) 161–175
3. L. Song and D. Hatzinakos,: Dense Wireless Sensor Networks with Mobile Sinks, Acoustics, Speech, and Signal Processing, Proceeding (2005) vol. 3, 677–680
4. Wang, Z.M. Basagni, S. Melachrinoudis, E. Petrioli, C.,: Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime, 38th Annual Hawaii International Conference on (2005) 287a
5. C. E. Perkins, E. M. Belding-Royer and S. Das,: Ad Hoc On Demand Distance Vector (AODV) routing, IETF Internet draft, (2002) 38
6. C. Intanagonwiwat, R. Govindan and D. Estrin,: Directed diffusion: A scalable and robust communication paradigm for sensor networks, Proc. ACM MOBICOM Conf. Boston, Massachusetts (2000) 56–67
7. http://www.isi.edu/nsnam.

Concurrent Diagnosis of Clustered Sensor Networks

Chin-Woo Cho and Yoon-Hwa Choi

Department of Computer Engineering,
Hongik University, 121-791 Seoul, Korea

Abstract. In this paper, we present an energy-efficient on-line diagnosis algorithm for cluster-based wireless sensor networks. It employs local comparisons of sensed data and dissemination of the decision made by the comparison results. Cluster-heads act as checkers for their associated cluster members. Redundant sensor nodes, as far as sensing coverage is concerned, are partially utilized to tolerate misbehavior of cluster-heads. Final decision on the fault status of sensor nodes is made at the base station. Computer simulation shows that high fault coverage can be achieved for a wide range of fault rates.

1 Introduction

Recently wireless sensor networks are emerging as computing platforms for various applications such as environmental monitoring, security surveillance, and target tracking [1]. Low-cost, low-power, tiny sensor nodes, which consist of sensing, processing, and communication units, are deployed to gather information from the environment and to deliver messages to a remote base station. Faults in sensor networks, however, may jeopardize the integrity of the sensor networks. Fault detection and diagnosis of wireless sensor networks for some particular applications has recently been investigated in [2][3][4]. In [3] each sensor node broadcasts its sensed data to its neighbors and compares its own data with the neighbors to recognize faults in an event region detection application. In [4] a cross-validation-based technique for detecting sensor faults has been developed.

In this paper, we present a diagnosis algorithm for clustered sensor networks [5]. It does not assume any fault-free diagnostic units in sensor nodes. Spare nodes are utilized as checker nodes for their associated clusters. Fault status of sensor nodes is determined during normal operation by combined efforts of cluster-heads (including the associated checker nodes) and the base station. Both sensing and computing faults are identified. Some communication faults may be covered by treating them as computing or sensing faults.

2 Clustered Structure for Diagnosis

In the diagnosis, we use comparisons of sensed data. A cluster-based sensor network, shown in Fig. 1, is used as our network model. Originally each cluster-head

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1267–1272, 2006.

at level 1 is expected to receive sensed data from its member nodes, performs data aggregation or fusion, and then sends the aggregated data to the base station. Each cluster-head checks to see if the sensing nodes are working correctly. A fault in the cluster-head, however, may render the diagnosis useless. To cope with this problem, some spare nodes, as far as sensing coverage is concerned, are temporarily used to do the same data aggregation and diagnosis. Nodes at level 1 are connected in Fig. 1 to indicate that they communicate with each other for diagnosis. Final decision on the fault status of sensor nodes will be made at the base station by collecting the diagnosis results and aggregated data from the cluster-heads and from the checker nodes, if necessary. The two-level hierarchy can be extended to even a higher-level structure, as long as the cluster size is not too small, without modifying the diagnosis algorithm to be presented shortly.

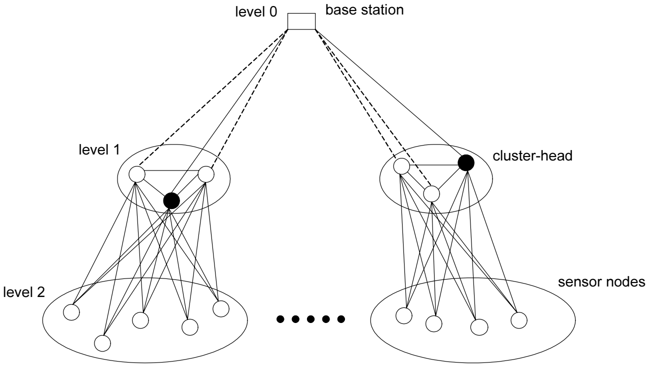


Fig. 1. Two-level hierarchy for fault diagnosis of sensor networks

3 Fault Model

The following fault model is used in the diagnosis. Faults may occur in any nodes in sensor networks, regardless of their locations (i.e., sensing, computing, communication units). Multiple faults may occur, although we assume that in a given cluster there is a single faulty level-1 node. This assumption can be easily removed if sufficient checker nodes (at least k+2 nodes for k faulty nodes) are employed. Also MTBF (mean-time-between-failure) is expected to be much longer than the diagnosis interval.

We also define the data model of the sensor networks, where a sensor node v is called a neighbor of a sensor node u if the distance between them is less than the sensing range. Let u and v be neighbors of each other and $s(u)$ denote the sensed data at node u. Then the condition to be satisfied by u and v is $|s(u) - s(v)| \leq \delta$, where δ may vary depending on the applications. In addition, an event may always be detected by more than $k_{event}(\geq 1)$ sensor nodes.

4 Fault Diagnosis Algorithm

A sensor network is represented here as a graph $G(U,E)$, where U represents the set of sensor nodes in the network and E represents the set of edges connecting sensor nodes. Two nodes u and v are said to be connected for diagnosis if the distance $d(u,v)$ is less than r (radius, sensing range, etc). $G(U,E)$ can also be called a test graph since u and v are compared only if $(u,v)\in E$. The comparison output is 0 if they satisfy the condition provided.

Definition 1: For the graph $G(U,E)$ and $u\in U$, the neighbors of u, $R(u)$ is defined to be $R(u) = \{v \in U : (u, v) \in E\}$.

Definition 2: For the graph $G(U,E)$, a label associated with $(u,v)\in E$ is represented as $S_u[v]$ and is a 0 if they satisfy the required condition addressed in the previous section. Otherwise, $S_u[v]$=1.

Definition 3: For the graph $G(U,E)$ and $u\in U$, the number of 0's in S_u is represented as $|S_u^0|$. Thus, $0 \leq |S_u^0| \leq |R(u)|$.

The diagnosis consists of three phases. It begins with a neighbor table NT, sorted in non-increasing order of node degrees, in each cluster-head-level node (cluster-head v_h and checker nodes v_c's). Each row of the table has received data, fault status indicator FSI (intialized to 1 (faulty)), a list of neighbors, and a flag V indicating whether the node has been visited or not.

Phase 1 (Diagnosing sensing nodes)**:** In this phase, v_h (also v_c's) determines fault-free nodes based on the following two decision criteria:

- For each member node u_i in the cluster, if $|S_i^0|\geq q$(threshold), then FSI_i is set to 0 (fault-free)
- For each member node u_i in the cluster, if it has a neighbor u_k such that FSI$_k$=0 and $|s(u_i) - s(u_k)| \leq \delta$, then u_i is determined to be fault-free.

The proposed diagnosis algorithm at phase 1 is depicted in Fig. 2. Depth-first search is used to visit the sensor nodes in NT until there are no neighbor nodes to visit.

If v_h is faulty, however, the decision made by v_h might be incorrect. Moreover, reliable fusion or aggregation of sensed data, coming from the member nodes, cannot be guaranteed.

```
Fault Diagnosis()
  Create a neighbor-table $NT$ (sorted)
  Initialize all $FSI$'s to 1(faulty) and $V$'s to 0 (not visited)
  While (until the last node of $NT$ with $V_i$=0)
  If $|S_i^0| \geq q$ then $FSI_i \leftarrow 0$; $V_i \leftarrow 1$ (fault-free); Propagation($i$);
            else if $|S_i^0|=0$, $V_i \leftarrow 1$ (faulty)
 Propagation ($i$)
   For each neighbor node $u_k$ with $S_i[k]=0$ and $V_k=0$
     $FSI_k \leftarrow 0$; $V_k \leftarrow 0$; Propagation($k$);
```

Fig. 2. Fault diagnosis for phase 1 in level-1 nodes

Phase 2 (Diagnosing cluster-heads)**:** The cluster head v_h and w checker nodes exchange their aggregated data and the results of phase 1 (FSI's) among themselves. During this process, the base station also receives the same aggregated data and the results of phase 1 from the cluster-head v_h. Each of them performs w comparisons (its own data with those of w other nodes) to see if v_h is fault-free. If the number of matches is greater than or equal to t (threshold), it determines itself fault-free. Each checker node, determined to be fault-free, sends its own FSI's and aggregated data it has generated to the base station only if it finds that v_h is faulty.

Under the given fault model, we need to consider the following two cases.

1) The cluster-head v_h is faulty. In this case (left side of Fig. 3), each of the checker nodes, $v'_c s$, will determine that v_h is faulty and send its own diagnosis results FSI's and and fused data to the base station. The dotted arrow means that the faulty cluster-head might send its own data to the base station.
2) One of the checker nodes is faulty. In this case (right side of Fig. 3), v_h finds itself fault-free and identifies the faulty checker node. Fault-free checker nodes, in the figure only one node v_{c1}, find that v_h is fault-free and will not send the diagnosis results to the base station. The faulty checker node, however, might send its own results to the base station, as indicated by a dotted arrow. The base station simply ignores the information if no other checker node reports that the cluster-head is faulty.

Phase 3 (Final decision at the base station)**:** The base station will receive FSI's and fused data from either the cluster-head (case 2) or more than one checker node (case 1). Based on the information, the fault status of each sensor node will be determined. In this process, an event has to be distinguished from a fault since both may assume sensing values outside of allowed range. In the case of an event, the diagnosis results are very similar to those of common mode failures. Hence the base station, which receives information from all the clusters, can distinguish events from faults, if common mode failures are unlikely to occur or can be controlled.

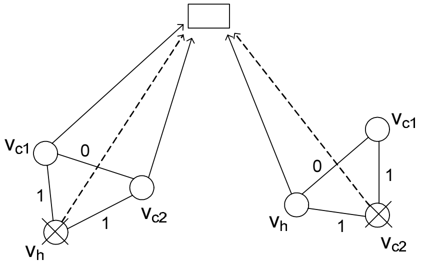


Fig. 3. Two possible fault patterns in level-1 nodes

5 Performance Evaluation

The performance of the proposed diagnosis algorithm is evaluated by computer simulation. Faults are assumed to be independent of each other. Also a sensor node is assumed to be faulty with probability p regardless of the location of the fault.

Let n_g and n_f be the numbers of fault-free nodes and faulty nodes in a sensor network, respectively. Also let n_{gg} be the number of fault-free nodes identified correctly and n_{fg} be the number of faulty nodes diagnosed as fault-free. The following two measures, $\alpha = \frac{\sum_{i=1}^{n}(n_{gg}/n_g)_i}{n}$ and $\beta = \frac{\sum_{i=1}^{n}(n_{fg}/n_f)_i}{n}$ are used to evaluate the performance, where n is the sample size. Apparently α and β lie in between 0 and 1 and our goal is to make α and β very close to one and zero, respectively.

In the simulation, nodes are placed randomly in an $l \times l$ (l=50m) rectangular region and 1000 sample clusters, randomly partitioned, are used to derive statistical data. For a given region with m sensor nodes, the average node degree ($\tilde{d}$) increases almost linearly with k (up to 8), where $k = \frac{m(\pi r^2)}{l^2}$ and r is the sensing range. The sensing coverage for k=3 is 0.906 and the corresponding $\tilde{d}$ is 2.26. Considering the fact that sensor networks need to maintain high coverage, the desired value of k would be greater than 3.

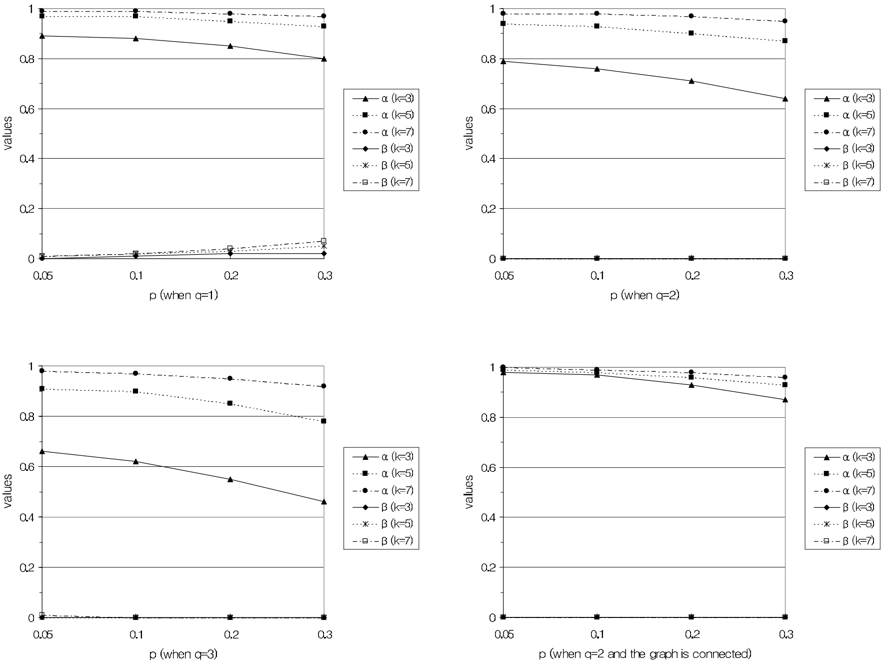


Fig. 4. α and β for various values of p, k, and q

Fig. 4 shows α and β for various values of p, k and q. As expected, α becomes very close to 1 as k (or $\tilde{d}$) increases even for a high p. At the same time, β can be almost perfectly controlled by properly choosing the value of q (threshold). As q increases, however, more fault-free nodes are likely to be misdiagnosed. Removing common mode failures by using higher threshold ends up with losing an increased number of fault-free nodes. Especially sensor nodes at the corners (or borderline) are at risk. Inter-cluster checking at the base station could enhance the performance since fault-free nodes isolated by faulty node(s) due to lower connectivity can be identified. The same simulation has been performed only for the test graphs connected. The results are shown in Fig. 4(d). As expected, a notable difference has been observed.

The proposed diagnosis algorithm is well suited with cluster-based communication. The overhead required is the energy consumed for exchanging diagnosis results and fused data among the cluster-head and checker nodes. Since w (number of checker nodes) is expected to be small, we can claim that the energy used for these communications is manageably small. Periodic checking can further reduce the energy consumed for diagnosis.

6 Conclusions

In this paper, we have presented a technique for locating faulty sensor nodes during normal operation. Faulty nodes have been identified without any fault-free diagnostic units. The diagnosis algorithm is well suited with the cluster-based communication protocols since most of the required communications occur through the paths originally established. As a result, the energy consumed for diagnosis has been minimized.

References

1. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci, "Wireless Sensor Networks: A Survey," Computer Networks, vol. 38, no.4, pp. 393-422, 2002.
2. C. Jaikaeo, C.Srisathapornphat, C-C. Shen, "Diagnosis of sensor networks," Int. Conf. Communications, vol 5, pp. 1627-1632, June 2001.
3. B. Krishnamachari, and S. Iyengar, "Bayesian Algorithms for Fault-tolerant Event Region Detection in Wireless Sensor Networks," IEEE Transactions on Computers, Vol. 53, No. 3, March 2004.
4. F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentelli, "On-line fault detection of sensor measurements," IEEE Sensors, vol. 2, pp. 974-980, Oct. 2003.
5. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660-670, Oct. 2002.

Author Index