

Lecture Notes in Computer Science

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

2093

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Singapore

Tokyo

Pascal Lorenz (Ed.)

Networking – ICN 2001

First International Conference on Networking
Colmar, France, July 9-13, 2001
Proceedings, Part I



Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Pascal Lorenz
University of Haute Alsace
IUT de Colmar – Departement GTR
34 rue du Grillenbreit
68008 Colmar, France
E-mail: lorenz@ieee.org

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Networking : first international conference ; proceedings / ICN 2001,
Colmar, France, July 9 - 13, 2001. Pascal Lorenz (ed.). - Berlin ;
Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ;
Singapore ; Tokyo : Springer
Pt. 1 . - (2001)
(Lecture notes in computer science ; Vol. 2093)
ISBN 3-540-42302-8

CR Subject Classification (1998): C.2, K.4.4, H.4.3, H.5.1, H.3, K.6.4-5

ISSN 0302-9743

ISBN 3-540-42302-8 Springer-Verlag Berlin Heidelberg New York

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

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2001
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Steingraber Satztechnik GmbH, Heidelberg
Printed on acid-free paper SPIN 10839401 06/3142 5 4 3 2 1 0

Preface

The International Conference on Networking (ICN01) is the first conference in its series aimed at stimulating technical exchange in the emerging and important field of networking. On behalf of the International Advisory Committee, it is our great pleasure to welcome you to the International Conference on Networking. Integration of fixed and portable wireless access into IP and ATM networks presents a cost effective and efficient way to provide seamless end-to-end connectivity and ubiquitous access in a market where demands on Mobile and Cellular Networks have grown rapidly and predicted to generate billions of dollars in revenue. The deployment of broadband IP - based technologies over Dense Wavelength Division Multiplexing (DWDM) and integration of IP with broadband wireless access networks (BWANs) are becoming increasingly important. In addition, fixed core IP/ATM networks are constructed with recent move to IP/MPLS over DWDM. Moreover, mobility introduces further challenges in the area that have neither been fully understood nor resolved in the preceding network generation. This first Conference ICN01 has been very well perceived by the International networking community. A total of 300 papers from 39 countries were submitted, from which 168 have been accepted. Each paper has been reviewed by several members of the scientific Program Committee.

The program covers a variety of research topics which are of current interest, such as mobile and wireless networks, Internet, traffic control, QoS, switching techniques, Voice over IP (VoIP), optical networks, Differentiated and Integrated services, IP and ATM networks, routing techniques, multicasting and performance evaluation, testing and simulation and modeling. Together with four tutorials and four Keynote Speeches, these technical presentations will address the latest research results from the international industries and academia and reports on findings from mobile, satellite and personal communications on 3rd and 4th generation research projects and standardization.

We would like to thank the scientific program committee members and the referees. Without their support, the program organization of this conference would not have been possible. We are also indebted to many individuals and organizations that made this conference possible (Association "Colmar-Liberty", GdR CNRS ARP, Ministère de la Recherche, Université de Haute Alsace, Ville de Colmar, France Telecom, IEEE, IEE, IST, WSES). In particular, we thank the members of the Organizing Committee for their help in all aspects of the organization of this conference.

We wish that you will enjoy this International Conference on Networking at Colmar, France and that you will find it a useful forum for the exchange of ideas and results and recent findings. We also hope that you will be able to spend some times to visit Colmar, with its beautiful countryside and its major cultural attractions.

General Chair
Pascal LORENZ
University of Haute Alsace
IUT de Colmar, France
lorenz@ieee.org

Technical Program Chair
Guy OMIDYAR
Center for Wireless Communications
National University of Singapore, SG
and Computer Sciences Corporation, USA
gomidyar@cw.c.nus.edu.sg

International Scientific Committee

H. Abouaissa (France) - University of Haute Alsace
R. Addie (Australia) - University of Southern Queensland
K. Begain (UK) - University of Bradford
A. Benslimane (France) - University of Belfort-Montbéliard
B. Bing (Singapore) - Ngee Ann Polytechnic
A. Brandwajn (USA) - University of California Santa Cruz
J.P. Coudreuse (France) - Mitsubishi
J. Crowcroft (UK) - University College London
S. Fdida (France) - LIP6
E. Fulp (USA) - Wake Forest University
B. Gavish (USA) - Vanderbilt University
H. Guyennet (France) - University of Franche-Comte
Z. Hulicki (Poland) - University of Cracow
R. Israel (France) - IEEE
A. Jajszczyk (Poland) - University of Mining & Metallurgy
A. Jamalipour (Australia) - University of Sydney
S. Jiang (Singapore) - National University of Singapore
S. Karnouskos (Germany) - GMD FOKUS
G. Kesidis (USA) - Pennsylvania State University
D. Khotimsky (USA) - Lucent Bell Labs
D. Kofman (France) - ENST Paris
S. Kota (USA) - Lockheed Martin
D. Kouvatsos (UK) - University of Bradford
S. Kumar (USA) - Ericsson
G.S. Kuo (Taiwan) - National Central University
F. Le Faucheur (France) - Cisco
M. Lee (Korea) - Dongshin University
P. Lorenz (France) - University of Haute Alsace
Z. Mammeri (France) - University of Toulouse
N. Mastorakis (Greece) - Military Institutions of University Education
H. Mouftah (Canada) - Queen's University
G. Omidyar (USA) - Computer Sciences Corp. (Program Chair)
J.J. Pansiot (France) - University of Strasbourg
M. Potts (Switzerland) - Martel
G. Pujolle (France) - University of Versailles-Saint-Quentin
S. Rao (Switzerland) - Ascom
A. Reid (UK) - British Telecom
S. Ritzenthaler (France) - Alcatel
P. Rolin (France) - ENST Bretagne
D. Sadok (Brazil) - Federal University of Pernambuco
R. Saracco (Italy) - CSELT
G. Swallow (USA) - Cisco
H. Tobiet (France) - Clemessy
M. Trehel (France) - University of Franche-Comte

VIII Organization

V.A. Villagra (Spain) - University of Madrid

E. Vazquez Gallo (Spain) - University of Madrid

O. Yang (Canada) - University of Ottawa

Table of Contents Part I

3rd to 4th Generation

Bandwidth Management for QoS Support in Mobile Networks	1
<i>S.-H. Lee, D.-S. Jung, Electronics & Telecommunication Research Institute, Korea ; S.-W. Park, Hannam University, Korea</i>	
3G and Beyond & Enabled Adaptive Mobile Multimedia Communication	12
<i>T. Kanter, Ericsson Radio Systems AB, Sweden ; T. Rindborg, D. Sahlin, Ericsson Utvecklings, Sweden</i>	
Creation of 3 rd Generation Services in the Context of Virtual Home Environment	27
<i>J. Oliveira, University of Porto, Portugal ; R. Roque, Portugal Telekom, Portugal ; S. Sedillot, INRIA, France ; E. Carrapatoso, University of Porto, Portugal</i>	

Internet (I)

WLL Link Layer Protocol for QoS Support of Multi-service Internet	37
<i>H. Pham, B. Lavery, James Cook University, Australia ; H.N. Nguyen, Vienna University of Technology, Austria</i>	
The Earliest Deadline First Scheduling with Active Buffer Management for Real-Time Traffic in the Internet	45
<i>X. Hei, D.H.K. Tsang, Hong Kong University of Science and Technology, Hong Kong</i>	
Pricing and Provisioning for Guaranteed Internet Services	55
<i>Z. Fan, University of Birmingham, UK</i>	
Price Optimization of Contents Delivery Systems with Priority	65
<i>K. Yamori, Y. Tanaka, Waseda University, Japan ; H. Akimaru, Asahi University, Japan</i>	
An Approach to Internet-Based Virtual Call Center Implementation	75
<i>M. Popovic, V. Kovacevic, University of Novi Sad, Yugoslavia</i>	

Traffic Control

Implementation and Characterization of an Advanced Scheduler	85
<i>F. Risso, Politecnico of Torino, Italy</i>	

A Performance Study of Explicit Congestion Notification (ECN) with Heterogeneous TCP Flows	98
<i>R. Kinicki, Z. Zheng, Worcester Polytechnic Institute, USA</i>	

Diffusion Model of RED Control Mechanism	107
<i>R. Laalaoua, S. Jedruś, T. Atmaca, T. Czachórski, INT, France</i>	

A Simple Admission Control Algorithm for IP Networks	117
<i>K. Kim, P. Mouchtaris, S. Samtani, R. Talpade, L. Wong, Telcordia Technologies, USA</i>	

An Architecture for a Scalable Broadband IP Services Switch	124
<i>M.V. Hegde, M. Naraghi-Poor, J. Bordes, C. Davis, O. Schmid, M. Maher, Celox Networks Inc., USA</i>	

Mobile and Wireless IP

On Implementation of Logical Time in Distributed Systems Operating over a Wireless IP Network	137
<i>D.A. Khotimsky, Lucent Technologies, USA ; I.A. Zhuklinets, Mozhaysky University, Russia</i>	

Performance of an Inter-segment Handover Protocol in an IP-based Terrestrial/Satellite Mobile Communications Network	147
<i>L. Fan, M.E. Woodward, J.G. Gardiner, University of Bradford, UK</i>	

Performance Evaluation of Voice-Data Integration for Wireless Data Networking	157
<i>M. Chatterjee, S.K. Das, The University of Texas at Arlington, USA ; G.D. Mandyam, Nokia Research Center, USA</i>	

A Hard Handover Control Scheme Supporting IP Host Mobility	167
<i>Y. Takahashi, N. Shinagawa, T. Kobayashi, YRP Mobile Telecommunications Key Technology Research Laboratories, Japan</i>	

Internet (II)

A Flexible User Authentication Scheme for Multi-server Internet Services	174
<i>W.-J. Tsaur, Da-Yeh University, Taiwan</i>	

NetLets: Measurement-Based Routing for End-to-End Performance over the Internet	184
<i>N.S.V. Rao, Oak Ridge National Laboratory, USA ; S. Radhakrishnan, B.-Y. Choel, University of Oklahoma, USA</i>	

Using the Internet in Transport Logistics – The Example of a Track & Trace System	194
<i>K. Jakobs, C. Pils, M. Wallbaum, Technical University of Aachen, Germany</i>	
Distributed Management of High-Layer Protocols and Network Services through a Programmable Agent-Based Architecture	204
<i>L.P. Gaspary, L.F. Balbinot, R. Storch, F. Wendt, L. Rockenbach Tarouco, Federal University of Rio Grande do Sul, Brazil</i>	
Differentiated Services	
A Buffer-Management Scheme for Bandwidth and Delay Differentiation Using a Virtual Scheduler	218
<i>R. Pletka, P. Droz, IBM Research, Switzerland ; B. Stiller, ETH Zürich, Switzerland</i>	
Enhanced End-System Support for Multimedia Transmission over the Differentiated Services Network	235
<i>H. Wu, Tsinghua University, China ; H.-R. Shao, Microsoft Research, China ; X. Li, Tsinghua University, China</i>	
Investigations into the Per-Hop Behaviors of DiffServ Networks	245
<i>Z. Di, H.T. Mouftah, Queen's University, Canada</i>	
An Adaptive Bandwidth Scheduling for Throughput and Delay Differentiation <i>H.-T. Nguyen, H. Rzehak, University of Federal Armed Forces Munich, Germany</i>	256
Evaluation of an Algorithm for Dynamic Resource Distribution in a Differentiated Services Network	266
<i>E.G. Nikolouzou, P.D. Sampatakis, I.S. Venieris, National Technical University of Athens, Greece</i>	
GPRS and Cellular Networks	
Quality of Service Management in GPRS Networks	276
<i>P. Stuckmann, F. Müller, Aachen University of Technology, Germany</i>	
Case Studies and Results on the Introduction of GPRS in Legacy Cellular Infrastructures	286
<i>C. Konstantinopoulou, K. Koutsopoulos, P. Demestichas, E. Matsikoudis, M. Theologou, National Technical University of Athens, Greece</i>	
Traffic Analysis of Multimedia Services in Broadband Cellular Networks	296
<i>P. Fazekas, S. Imre, Budapest University of Technology and Economics, Hungary</i>	

Scheduling Disciplines in Cellular Data Services with Probabilistic Location Errors	307
<i>J.-L. Chen, H.-C. Cheng, H.-C. Chao, National Dong Hwa University, Taiwan</i>	

WDM and Optical Networks

Restoration from Multiple Faults in WDM Networks without Wavelength Conversion	317
<i>C.-C. Sue, S.-Y. Kuo, National Taiwan University, Taiwan</i>	

An All-Optical WDM Packet-Switched Network Architecture with Support for Group Communication	326
<i>M.R. Salvador, S.Heemstra de Groot, D. Dey, University of Twente, The Netherlands</i>	

Performance Consideration for Building the Next Generation Multi-service Optical Communications Platforms	336
<i>S. Dastangoo, Onex™ Communications Corporation, USA</i>	

Traffic Management in Multi-service Optical Network	348
<i>H. Elbiaze, T. Atmaca, INT, France</i>	

Performance Comparison of Wavelength Routing Optical Networks with Chordal Ring and Mesh-Torus Topologies	358
<i>M.M. Freire, University of Beira Interior, Portugal ; H.J.A. da Silva, University of Coimbra, Portugal</i>	

Differentiated and Integrated Services

Achieving End-to-End Throughput Guarantee for Individual TCP Flows in a Differentiated Services Network	368
<i>X. He, H. Che, The Pennsylvania State University, USA</i>	

Performance Evaluation of Integrated Services in Local Area Networks	378
<i>J. Ehrensberger, Swiss Federal Institute of Technology, Switzerland</i>	

Integrating Differentiated Services with ATM	388
<i>S. Manjanatha, R. Bartoš, University of New Hampshire, USA</i>	

Management and Realization of SLA for Providing Network QoS	398
<i>M. Hashmani, M. Yoshida, NS Solutions Corporation, Japan ; T. Ikenaga, Y. Oie, Kyushu Institute of Technology, Japan</i>	

Optimal Provisioning and Pricing of Internet Differentiated Services in Hierarchical Markets	409
<i>E.W. Fulp, Wake Forest University, USA ; D.S. Reeves, N.C. State University, USA</i>	
Keynote Speech	
Adding Interactive Services in a Video Broadcasting Network	419
<i>R. Jäger, BetaResearch, Germany</i>	
Wireless ATM	
A Discrete-Time Queuing Analysis of the Wireless ATM Multiplexing System	429
<i>M.M. Ali, X. Zhang, J.F. Hayes, Concordia University, Canada</i>	
A QoS Based Distributed Method for Resource Allocation in Unlicensed Wireless ATM Systems	439
<i>G.F. Marias, L. Merakos, University of Athens, Greece</i>	
An Adaptive Error Control Mechanism for Wireless ATM	449
<i>P.R. Denz, A.A. Nilsson, North Carolina State University, USA</i>	
A Review of Call Admission Control Schemes in Wireless ATM Networks	459
<i>D.D. Vergados, N.G. Protopsaltis, C. Anagnostopoulos, J. Anagnostopoulos, M.E. Theologou, E.N. Protonotarios, National Technical University of Athens, Greece</i>	
Multicast (I)	
Analysis and Evaluation of QoS-Sensitive Multicast Routing Policies	468
<i>E. Pagani, G.P. Rossi, University of Milano, Italy</i>	
Multicast Performance of Multistage Interconnection Networks with Shared Buffering	478
<i>D. Tutsch, International Computer Science Institute, USA ; M. Hendler, G. Hommel, Technical University of Berlin, Germany</i>	
Performance Evaluation of PIM-SM Recovery	488
<i>T. Čičić, S. Gjessing, University of Oslo, Norway ; Ø. Kure, Norwegian University of Science and Technology, Norway</i>	
Reducing Multicast Inter-receiver Delay Jitter – A Server Based Approach	498
<i>J.-U. Klöcking, C. Maihöfer, K. Rothermel, University of Stuttgart, Germany</i>	
Multicast Routing and Wavelength Assignment in Multi-Hop Optical Networks	508
<i>R. Libeskind-Hadas, Harvey Mudd College, USA ; R. Melhem, University of Pittsburgh, USA</i>	

Real-Time Traffic

Improving the Timed Token Protocol	520
<i>Y. Bouzida, R. Beghdad, EMP, Algeria</i>	
Design of a Specification Language and Real-Time APIs for Easy Expression of Soft Real-Time Constraints with Java	530
<i>K.-Y. Sung, Handong University, Korea</i>	
Feedback-Controlled Traffic Shaping for Multimedia Transmissions in a Real-Time Client-Server System	540
<i>G.-M. Muntean, L. Murphy, University College Dublin, Ireland</i>	
Bandwidth Reallocation Techniques for Admitting High Priority Real-Time Calls in ATM Networks	549
<i>L.K. Miller, University of Toledo, USA ; E.L. Leiss, University of Houston, USA</i>	
Temporal Control Specifications and Mechanisms for Multimedia Multicast Communication Services	559
<i>H.-Y. Kung, National Pingtung University of Science and Technology, Taiwan; C.-M. Huang, National Cheng Kung University, Taiwan</i>	

Wireless (I)

Improving Fairness and Throughput in Multi-Hop Wireless Networks	569
<i>H.-Y. Hsieh, R. Sivakumar, Georgia Institute of Technology, USA</i>	
Dynamic Allocation of Transmitter Power in a DS-CDMA Cellular System Using Genetic Algorithms	579
<i>J. Zhou, Y. Shiraishi, U. Yamamoto, Y. Onozato, Gunma University, Japan</i>	
The Coexistence of Multicast and Unicast over a GPS Capable Network	589
<i>T. Asfour, A. Serhrouchni, ENST, France</i>	
An Approach for QoS Scheduling on the Application Level for Wireless Networks	599
<i>D.-H. Hoang, D. Reschke, TU Ilmenau, Germany</i>	

Multicast (II)

A Host-Based Multicast (HBM) Solution for Group Communications	610
<i>V. Roca, A. El-Sayed, INRIA Rhône-Alpes, France</i>	
An Evaluation of Shared Multicast Trees with Multiple Active Cores	620
<i>D. Zappala, A. Fabbri, University of Oregon, USA</i>	

QoS Routing Protocol for the Generalized Multicast Routing Problem (GMRP) <i>H. Bettahar, A. Bouabdallah, Technical University of Compiègne, France</i>	630
Feedback Scalability for Multicast Videoconferencing..... <i>H. Smith, California Polytechnic State University, USA ; M. Mutka, L. Yang, Michigan State University, USA</i>	640
Group Communication and Multicast <i>J. Templemore-Finlayson, S. Budkowski, INT, France</i>	649
Routing	
Avoiding Counting to Infinity in Distance Vector Routing <i>A. Schmid, O. Kandel, C. Steigner, University of Koblenz-Landau, Germany</i>	657
Proposal of an Inter-AS Policy Routing and a Flow Pricing Scheme to Improve ASes' Profits <i>N. Ogino, M. Suzuki, KDD R&D Laboratories Inc., Japan</i>	673
Stigmergic Techniques for Solving Multi-constraint Routing for Packet Networks <i>T. Michalareas, L. Sacks, University College London, UK</i>	687
Dynamic Capacity Resizing of Virtual Backbone Networks <i>S.H. Rhee, Kwangwoon University, Korea ; J. Yoon, H. Choi, I. Choi, ETRI/National Security Research Institute, Korea</i>	698
Wireless (II)	
Throughput Improvements Using the Random Leader Technique for the Reliable Multicast Wireless LANs <i>H.-C. Chao, National Dong Hwa University, Taiwan ; S.W. Chang, Southern Information System Inc., Taiwan ; J.L. Chen, National Dong Hwa University, Taiwan</i>	709
Performance Evaluation and Implementation of QoS Support in an 802.11b Wireless LAN <i>V. Mirchandani, E. Dutkiewicz, Motorola Australian Research Centre, Australia</i>	720
Increasing Throughput and QoS in a HIPERLAN/2 System with Co-channel Interference <i>J. Rapp, Aachen University of Technology, Germany</i>	727
An Intra-media Multimode Wireless Communication Terminal for DSRC Service Networks <i>M. Umemoto, Yokosuka ITS Research Center, Japan</i>	737

Traffic Analysis, Modeling and Simulation

Simulation of Traffic Engineering in IP-Based Networks	743
<i>T.T.M. Hoang, W. Zorn, University of Karlsruhe, Germany</i>	
An Algorithm for Available Bandwidth Measurement	753
<i>J. He, C.E. Chow, J. Yang, T. Chujo, Fujitsu Laboratories of America Inc., USA</i>	
Influence of Network Topology on Protocol Simulation	762
<i>D. Magoni, J.-J. Pansiot, Université Louis Pasteur, France</i>	
An Adaptive Flow-Level Load Control Scheme for Multipath Forwarding	771
<i>Y. Lee, Y. Choi, Seoul National University, Korea</i>	
Performance Evaluation of CANIT Algorithm in Presence of Congestion Losses	780
<i>H. Benaboud, N. Mikou, University of Bourgogne, France</i>	

User Applications

Project Driven Graduate Network Education	790
<i>A. Van de Capelle, E. Van Lil, J. Theunis, J. Potemans, M. Teughels, K.U. Leuven, Belgium</i>	
Modelling User Interaction with E-commerce Servers	803
<i>H. Graja, J. McManis, School of Electronic Engineering, Ireland</i>	
XML Smartcards	811
<i>P. Urien, H. Saleh, A. Tizraoui, Bull, France</i>	
The Influence of Web Page Images on the Performance of Web Servers	821
<i>C. Hava Muntean, J. McManis, J. Murphy, Dublin City University, Ireland</i>	
Network Resilience in Multilayer Networks: A Critical Review and Open Issues	829
<i>F. Touvet, D. Harle, University of Strathclyde, UK</i>	
Author Index	839

Table of Contents Part II

Mobility Management

Dynamic Resource Management Scheme for Multimedia Services in Wireless Communication Networks	1
<i>D.-E. Lee, Chungwoon University, Korea ; B.-J. Lee, J.-W. Ko, Y.-C. Kim, Chonbuk National University, Korea</i>	
An Adaptive Handover-Supporting Routing Method for ATM Based Mobile Networks, and Its Adaptation to IP Scenarios	11
<i>S. Szabo, S. Imre, Budapest University of Technology and Economics, Hungary</i>	
Location Stamps for Digital Signatures: A New Service for Mobile Telephone Networks	20
<i>M. Kabatnik, University of Stuttgart, Germany ; A. Zugenmaier, University of Freiburg, Germany</i>	
A New Method for Scalable and Reliable Multicast System for Mobile Networks	31
<i>M. Hayashi, Hitachi Europe, France ; C. Bonnet, Institute Eurecom, France</i>	
An Adaptive Mobility Management Scheme to Support Internet Host Mobility	41
<i>M. Woo, Sejong University, Korea</i>	

TCP Analysis

Modeling and Analysis of TCP Enhancement over Heterogeneous Links	51
<i>M. Liu, N. Ehsan, University of Michigan, USA</i>	
TCP Throughput Guarantee Using Packet Buffering	61
<i>S. Choi, C. Kim, Seoul National University, Korea</i>	
Modular TCP Handoff Design in STREAMS-Based TCP/IP Implementation ...	71
<i>W. Tang, L. Cherkasova, L. Russell, Hewlett Packard Labs., USA ; M.W. Mutka, Michigan State University, USA</i>	
An Efficient TCP Flow Control and Fast Recovery Scheme for Lossy Networks	82
<i>H.Y. Liao, Y.C. Chen, C.L. Lee, National Chiao Tung University, Taiwan</i>	
Bandwidth Tradeoff between TCP and Link-Level FEC	97
<i>C. Barakat, E. Altman, INRIA, France</i>	

QoS (I)

Supporting QoS for Legacy Applications	108
<i>C. Tsetsekas, S. Maniatis, I.S. Venieris, National Technical University of Athens, Greece</i>	
An Open Architecture for Evaluating Arbitrary Quality of Service Mechanisms in Software Routers	117
<i>K. Wehrle, University of Karlsruhe, Germany</i>	
Measurement-Based IP Transport Resource Manager Demonstrator	127
<i>V. Räisänen, Nokia Research Center, Finland</i>	
Packet-Size Based Queuing Algorithm for QoS Support	137
<i>M.C. Choi, H.L. Owen, Georgia Tech Research Institute, USA ; J. Sokol, Siemens AG, Germany</i>	
Backbone Network Design with QoS Requirements	148
<i>H.-H. Yen, F.Y.-S. Lin, National Taiwan University, Taiwan</i>	

Ad Hoc Networks

A Multi-path QoS Routing Protocol in a Wireless Mobile ad Hoc Network	158
<i>W.-H. Liao, National Central University, Taiwan ; Y.-C. Tseng, National Chiao-Tung University, Taiwan ; S.-L. Wang, J.-P. Sheu, National Central University, Taiwan</i>	
Study of a Unicast Query Mechanism for Dynamic Source Routing in Mobile ad Hoc Networks	168
<i>B.-C. Seet, B.-S. Lee, C.-T. Lau, Nanyang Technological University, Singapore</i>	
Ad-hoc Filesystem:	
A Novel Network Filesystem for Ad-hoc Wireless Networks	177
<i>K. Yasuda, T. Hagino, Keio University, Japan</i>	
A Review of Current On-demand Routing Protocols	186
<i>M. Abolhasan, T. Wysocki, University of Wollongong, Australia ; E. Dutkiewicz, Motorola Australia Research Centre, Australia</i>	

Security

Construction of Data Dependent Chaotic Permutation Hashes to Ensure Communications Integrity	196
<i>J. Scharinger, Johannes Kepler University, Austria</i>	

Secure Communication: A New Application for Active Networks	206
<i>M. Günter, M. Brogle, T. Braun, University of Berne, Switzerland</i>	
Deployment of Public-Key Infrastructure in Wireless Data Networks	217
<i>A.K. Singh, Infosys Technologies Limited, India</i>	
A Scalable Framework for Secure Group Communication	225
<i>L.-C. Wu, H.-C. Chen, National YunLin University, Taiwan</i>	
Authenticating Multicast Streams in Lossy Channels Using Threshold Techniques	239
<i>M. Al-Ibrahim, University of Wollongong, Australia ; J. Pieprzyk, Macquarie University, Australia</i>	
QoS (II)	
Tuning of QoS Aware Load Balancing Algorithm (QoS-LB) for Highly Loaded Server Clusters	250
<i>K. Kaario, Honeywell Industrial Automation & Control, Finland ; T. Hämäläinen, J. Zhang, University of Jyväskylä, Finland</i>	
The Incremental Deployability of Core-Stateless Fair Queuing	259
<i>Y. Blanpain, H.-Y. Hsieh, R. Sivakumar, Georgia Institute of Technology, USA</i>	
A New IP Multicast QoS Model on IP Based Networks	268
<i>H.S. Eissa, T. Kamel, Electronics Research Institute, Egypt</i>	
Integrated Management of QoS-Enable Networks Using QAME	277
<i>L. Zambenedetti Granville, L.M. Rockenbach Tarouco, M. Bartz Cecon, M.J. Bosquioli Almeida, Federal University of Rio Grande do Sul, Brazil</i>	
On Web Quality of Service: Approaches to Measurement of End-to-End Response Time	291
<i>M. Tsykin, Fujitsu Australia Limited, Australia</i>	
MPLS	
Path Computation for Traffic Engineering in MPLS Networks	302
<i>G. Banerjee, D. Sidhu, University of Maryland, USA</i>	
Minimum Regret Approach to Network Management under Uncertainty with Applications to Connection Admission Control and Routing	309
<i>V. Marbukh, National Institute of Standards and Technology, USA</i>	

MPLS Restoration Scheme Using Least-Cost Based Dynamic Backup Path	319
<i>G. Ahn, Electronics and Telecommunications Research Institute, Korea ; W. Chun, Chungnam National University, Korea</i>	
Connection Management in MPLS Networks Using Mobile Agents	329
<i>S. Yucel, Marconi Communications, USA ; T. Saydam, University of Delaware, USA</i>	
General Connection Blocking Bounds and an Implication of Billing for Provisioned Label-Switched Routes in an MPLS Internet Cloud	339
<i>G. Kesidis, Pennsylvania State University, USA ; L. Tassiulas, University of Maryland, USA</i>	
Switches	
FPCF Input-Queued Packet Switch for Variable-Size Packets	348
<i>P. Homan, J. Bester, University of Ljubljana, Slovenia</i>	
A Cost-Effective Hardware Link Scheduling Algorithm for the Multimedia Router (MMR)	358
<i>M.B. Caminero, C. Carrión, F.J. Quiles, University of Castilla-La Mancha, Spain ; J. Duato, Polytechnical University of Valencia, Spain ; S. Yalamanchili, Georgia Institute of Technology, USA</i>	
The Folded Hypercube ATM Switches	370
<i>J.S. Park, N.J. Davis IV, Virginia Polytechnic Institute and State University, USA</i>	
Open Software Architecture for Multiservice Switching System	380
<i>H.-J. Park, Y.-I. Choi, B.-S. Lee, K.-P. Jun, Electronics & Telecommunication Research Institute, Korea</i>	
A Multicast ATM Switch Based on PIPN	390
<i>S.F. Oktug, Istanbul Technical University, Turkey</i>	
CORBA	
Concurrent Access to Remote Instrumentation in CORBA-Based Distributed Environment	399
<i>A. Stranjak, Lucent Technologies, Ireland ; D. Kovačić, I. Čavrak, M. Žagar, University of Zagreb, Croatia</i>	
Design and Implementation of CORBA-Based Integrated Network Management System	409
<i>J.-H. Kwon, HyComm Incorporated, USA ; J.-T. Park, Kyungpook National University, Korea</i>	

Framework for Real-Time CORBA Development	422
<i>Z. Mammeri, J. Rodriguez, University of Toulouse, France ;</i> <i>P. Lorenz, IUT de Colmar, University of Haute Alsace, France</i>	
Development of Accounting Management Based Service Environment in Tina, Java and Corba Architectures	438
<i>A. Sekkaki, University Hassan II, Morocco ; L.M. Cáceres Alvarez,</i> <i>W. Tatsuya Watanabe, C. Becker Westphall, Federal University of Santa</i> <i>Catarina, Brazil</i>	
A QoS System for CaTV Networks	449
<i>J. Leal, J.M. Fornés, University of Sevilla, Spain</i>	
Mobile Agents	
Towards Manageable Mobile Agent Infrastructures	458
<i>P. Simões, P. Marques, L. Silva, J. Silva, F. Boavida, University of Coimbra,</i> <i>Portugal</i>	
Dynamic Agent Domains in Mobile Agent Based Network Management	468
<i>R. Sugar, S. Imre, Technical University of Budapest, Hungary</i>	
Networking in a Service Platform Based on Mobile Agents	478
<i>M. Palola, VTT Electronics, Finland</i>	
Realizing Distributed Intelligent Networks Based on Distributed Object and Mobile Agent Technologies	488
<i>M.K. Perdikeas, O.I. Pyrovolakis, A.E. Papadakis, I.S. Venieris, National</i> <i>Technical University of Athens, Greece</i>	
ATM Networks (I)	
A New Cut-Through Forwarding Mechanism for ATM Multipoint-to-Point Connections	497
<i>A. Papadopoulos, Computers Technology Institute, Greece ;</i> <i>T. Antonakopoulos, V. Makios, University of Patras, Greece</i>	
Cell-by-Cell Round Robin Service Discipline for ATM Networks	507
<i>H.M. Mokhtar, R. Pereira, M. Merabti, Liverpool John Moores University, UK</i>	
Delay and Departure Analysis of CBR Traffic in AAL MUX with Bursty Background Traffic	517
<i>C.G. Park, D.H. Han, Sunmoon University, Korea</i>	
Virtual Path Layout in ATM Path with Given Hop Count	527
<i>S. Choplin, INRIA, France</i>	

Simulation-Based Stability of a Representative, Large-Scale ATM Network for a Distributed Call Processing Architecture	538
<i>R. Citro, Intel Corporation, USA ; S. Ghosh, Stevens Institute of Technology, USA</i>	

Voice over IP (I)

Proposed Architectures for the Integration of H.323 and QoS over IP Networks	549
<i>R. Estepa, J. Leal, J.A. Ternerero, J.M. Vozmediano, University of Sevilla, Spain</i>	
Third-Party Call Control in H.323 Networks – A Case Study	559
<i>A. Miloslavski, V. Antonov, L. Yegoshin, S. Shkrabov, J. Boyle, G. Pogosyants, N. Anisimov, Genesys Telecommunication Labs, USA</i>	
Measurement-Based MMPP Modeling of Voice Traffic in Computer Networks Using Moments of Packet Interarrival Times	570
<i>N.S. Kambo, D.Z. Deniz, T. Iqbal, Eastern Mediterranean University, Turkey</i>	
Evaluation of End-to-End QoS Mechanisms in IP Networks	579
<i>F.A. Shaikh, S. McClellan, University of Alabama at Birmingham, USA</i>	
Web-Enabled Voice over IP Call Center	590
<i>S. Kuhlins, University of Mannheim, Germany ; D. Gutacker, OSI mbH, Germany</i>	

Active Networks

ANMP: Active Network Management Platform for Telecommunications Applications	599
<i>W.-K. Hong, M.-J. Jung, Korea Telecom, Korea</i>	
An Active Network Architecture: Distributed Computer or Transport Medium .	612
<i>E. Hladká, Z. Salvét, Masaryk University, Czech Republic</i>	
An Active Network for Improving Performance of Traffic Flow over Conventional ATM Service	620
<i>E. Rashid, T. Araki, Hirosaki University, Japan ; T. Nakamura, Tohoku University, Japan</i>	
An Active Programmable Harness for Measurement of Composite Network States	628
<i>J.I. Khan, A.U. Haque, Kent State University, USA</i>	
Protocol Design of MPEG-4 Media Delivery with Active Networks	639
<i>S. Go, J.W. Wong, University of Waterloo, Canada ; Z. Wu, Bond University, Australia</i>	

ATM Networks (II)

Prediction and Control of Short-Term Congestion in ATM Networks Using Artificial Intelligence Techniques	648
<i>G. Corral, A. Zaballos, J. Camps, J.M. Garrell, University Ramon Llull, Spain</i>	
Monitoring the Quality of Service on an ATM Network Carrying MMB Traffic Using Weighted Significance Data	658
<i>A.A.K. Mouharam, M.J. Tunncliffe, Kingston University, UK</i>	
ATM Traffic Prediction Using Artificial Neural Networks and Wavelet Transforms	668
<i>P. Solís Barreto, Catholic University of Goiás, Brazil ; R. Pinto Lemos, Federal University of Goiás, Brazil</i>	
Threshold-Based Connection Admission Control Scheme in ATM Networks: A Simulation Study	677
<i>X. Yuan, North Carolina A & T State University, USA ; M. Ilyas, Florida Atlantic University, USA</i>	
ABR Congestion Control in ATM Networks Using Neural Networks	687
<i>K. Dimyati, C.O. Chow, University of Malaya, Malaysia</i>	

Voice over IP (II)

An Architecture for the Transport of IP Telephony Services	697
<i>S. Guerra, J. Vinyes, D. Fernández, Technical University of Madrid, Spain</i>	
Architectural Framework for Using Java Servlets in a SIP Environment	707
<i>R. Glitho, R. Hamadi, R. Huie, Ericsson Research Canada, Canada</i>	
A Practical Solution for Delivering Voice over IP	717
<i>S. Milanovic, Serco Group plc, Italy; Z. Petrovic, University of Belgrade, Yugoslavia</i>	
QoS Guaranteed Voice Traffic Multiplexing Scheme over VoIP Network Using DiffServ	726
<i>E.-J. Ha, J.-H. Kwon, J.-T. Park, Kyungpook National University, Korea</i>	
VoIP over MPLS Networking Requirements	735
<i>J.-M. Chung, Oklahoma State University, USA ; E. Marroun, H. Sandhu, Cisco Systems, USA ; S.-C. Kim, Oklahoma State University, USA</i>	

Video Communications

A System Level Framework for Streaming 3-D Meshes over Packet Networks . <i>G. Al-Regib, Y. Altunbasak, Georgia Institute of Technology, USA</i>	745
Techniques to Improve Quality-of-Service in Video Communications via Best Effort Networks <i>B.E. Wolfinger, M. Zaddach, Hamburg University, Germany</i>	754
Simulation of a Video Surveillance Network Using Remote Intelligent Security Cameras <i>J.R. Renno, M.J. Tunnicliffe, G.A. Jones, Kingston University, UK ; D.J. Parish, Loughborough University, UK</i>	766
Cooperative Video Caching for Interactive and Scalable VoD Systems <i>E. Ishikawa, C. Amorim, Federal University of Rio de Janeiro, Brazil</i>	776
Optimal Dynamic Rate Shaping for Compressed Video Streaming <i>M. Kim, Y. Altunbasak, Georgia Institute of Technology, USA</i>	786

ATM Networks (III)

IP Stack Emulation over ATM <i>I.G. Goossens, I.M. Goossens, Free University of Brussels, Belgium</i>	795
ATM Network Restoration Using a Multiple Backup VPs Based Self-Healing Protocol <i>S.N. Ashraf, INT, France ; C. Lac, France Télécom R&D, France</i>	805
A New Consolidation Algorithm for Point-to-Multipoint ABR Service in ATM Networks <i>M. Shamsuzzaman, A.K. Gupta, B.-S. Lee, Nanyang Technological University, Singapore</i>	815
Experimental TCP Performance Evaluation on DiffServ Assured Forwarding over ATM SBR Service <i>S. Ano, T. Hasegawa, KDDI R&D Laboratories Inc, Japan ; N. Decre, ENST, France</i>	825
PMS: A PVC Management System for ATM Networks <i>C. Yang, S. Phan, National Research Council of Canada, Canada</i>	836

Modelization

Buffer-Size Approximation for the Geo/D/1/K Queue	845
<i>P. Linwong, Tohoku University, Japan ; A. Fujii, Miyagi University, Japan ; Y. Nemoto, Tohoku University, Japan</i>	
Client-Server Design Alternatives: Back to Pipes but with Threads	854
<i>B. Roussev, Susquehanna University, USA ; J. Wu, Florida Atlantic University, USA</i>	
Towards a Descriptive Approach to Model Adaptable Communication Environments	867
<i>A.T.A. Gomes, S. Colcher, L.F.G. Soares, PUC-Rio, Brazil</i>	
All-to-All Personalized Communication Algorithms in Chordal Ring Networks	877
<i>H. Masuyama, H. Taniguchi, T. Miyoshi, Tottori University, Japan</i>	
Formal and Practical Approach to Load Conditions in High Speed Networks ...	890
<i>A. Pollak, University of the Armed Forces Munich, Germany</i>	
Author Index	895

Bandwidth Management for QoS Support in Mobile Networks

Sang-Ho Lee¹, Dong-Soo Jung¹, and Sung-Woo Park²

¹Mobile Call Processing Team, Electronics and Telecommunication Research Institute,
161 Kajong-dong, Yusong-gu, Taejon, 305-600, Korea
{leesh, dsjung}@etri.re.kr

²Dept. of Information & Communication Engineering, Hannam University, Korea
swpark@eve.hannam.ac.kr

Abstract. A bandwidth management scheme is proposed that can support the seamless QoS in face of handoff in mobile networks. The proposed scheme is based on the time-selective bandwidth reservation with the reduced signaling and computational overhead. The reservation parameters are adjusted dynamically to cope with user mobility. Throughout the computer simulations, the performance of the proposed scheme is evaluated. The simulation results show that the handoff call blocking probability can be remarkably improved with a slight degradation of other parameters such as new call blocking probability and bandwidth utilization efficiency.

1 Introduction

The next generation high-speed mobile networks are expected to support multimedia applications. As such, it is important that these networks provide the quality-of-service (QoS) guarantees. QoS support for multimedia traffic has been extensively studied for wired networks. Supporting QoS in mobile networks is complicated due to user mobility and unreliable radio channels. This problem becomes even more challenging as recent mobile networks tend to be deployed over small-size cells (i.e., micro-cells or pico-cells) to allow higher transmission capacity.

One of the main objectives of IMT-2000 is to provide mobile users with multimedia services. Internet services also play an important role in IMT-2000 as the Internet grows rapidly. For data services to be implemented efficiently, a separate mobile packet network based on either GPRS or Mobile IP has been suggested. To support QoS in mobile packet networks, somewhat different service strategies need to be developed with the mobile's characteristics sufficiently taken into consideration [1].

First of all, the QoS architecture in mobile networks must be built on the top of the existing QoS concepts used in wired networks. Then, the effects of user mobility and wireless communication channels should be incorporated into the architecture. Especially, our focus is on the QoS features related to the user mobility. We call this 'mobile-QoS'. The mobile-QoS involves additional handoff-related parameters such as handoff blocking rate etc. To guarantee the mobile-QoS, bandwidth reservation is

conceived as one of the most assured schemes. When a mobile is located in a cell, the bandwidth of wireless channels in some neighbor cells are reserved in advance for the mobile's handoff. Unless properly managed, however, bandwidth reservation may incur a significant waste of network resources.

The existing bandwidth reservation schemes may be divided into two categories: static reservation and dynamic reservation [2]. The dynamic reservation scheme is classified into two types again: time-continuous reservation [3] and time-selective reservation [4][5]. The former reserves bandwidth on all neighbor cells since new calls were generated until they are terminated. But, in the latter case, bandwidth reservation is done selectively on the neighbor cells according to the estimated arrival time of the mobile at each cell. It is obvious that the latter shows better bandwidth utilization against the former, but signaling and computational overhead is increased.

The purpose of the QoS-based bandwidth management in mobile networks is to provide the requested QoS of each call regardless of mobile's handoff while maintaining the maximum utilization of network resources. To this end, this paper proposes a new bandwidth management scheme that can support the seamless mobile-QoS in the face of handoff in mobile networks. The proposed scheme is essentially based on the time-selective bandwidth reservation described in [4] and [5]. The difference is that the estimation of user mobility is done by the aggregated measurement instead of call-by-call computation. By this way, signaling and computational overheads can be significantly reduced.

This paper is organized as follows. In Section 2, we establish a framework for bandwidth reservation and a reservation model to support the mobile-QoS. In Section 3, we propose a reservation-based bandwidth management scheme. In Section 4, the performance of the proposed scheme is evaluated using computer simulations and some numerical results are provided. Finally, we conclude our work in Section 5.

2 Mobile-QoS Framework

2.1 Reservation Parameters

With regard to the bandwidth reservation for the mobile-QoS, we encounter three fundamental questions: where (the selection of neighbor cells to be reserved), when (the decision of starting time and ending time for reservation), and how much (the allocation of certain amount of bandwidth for reservation)? To answer these questions, we introduce the following parameters: reservation range, reservation interval, and reservation bandwidth.

Reservation Range. When a new call is generated in a particular cell, the reservation range represents the set of neighbor cells for which certain amount of bandwidth is reserved. The reservation range should be properly chosen depending on the requested QoS and the user mobility. If the reservation range is larger than required, bandwidth will be underutilized. Conversely, if it is less, then the requested QoS cannot be properly supported.

Reservation Interval. If a cell has been selected to be included in the reservation range, then the next thing to do is to decide the reservation interval on the time axis. To do this, both arrival time and departure time of the mobiles should be estimated as precisely as possible. Based on these estimations, the starting time and the ending time of reservation interval are determined.

The arrival time is the sum of residence time in each cell traversed by the mobile. The residence time is again a function of cell size and user mobility. The departure time also depends on the residence time in the cell. In this case, however, the call holding time must be also considered. If we want to offer better mobile-QoS, some guard times can be appended to both ends of the interval.

Reservation Bandwidth. Once the reservation interval has been set, the amount of reservation bandwidth is also an important factor that has direct impacts on the mobile-QoS. The reservation bandwidth is basically related to the required mobile-QoS. Unlike the fixed network, the reserved bandwidth would be proportional to the required bandwidth as well as the required handoff blocking probability.

Table 1 indicates the network or mobile characteristics that affect the reservation parameters.

Table 1. Reservation parameters

Reservation parameters Characteristics		Reservation range	Reservation interval		Reservation bandwidth
			Arrival time	Departure time	
Cell size		O	O	O	
Call holding time		O		O	
Mobile-QoS	BW				O
	Handoff blocking prob.	O	O	O	O
Mobility	Speed	O	O	O	
	Direction	O	O	O	

2.2 Design Principles

Recall that the objective of bandwidth management is to maximize the utilization of network resources and at the same time meet the QoS requirements. Then there naturally arises a trade-off between bandwidth utilization and the degree of mobile-QoS provided in determining the reservation parameters. From the viewpoint of implementation, additional important factor to be considered is the simplicity represented by signaling and computation.

The most straightforward way to support the mobile-QoS is overprovisioning. Now, if we estimate the user mobility more accurately, the same level of mobile-QoS could be supported with the less bandwidth required. However, this can be achieved

at the expense of the simplicity due to the increased signaling and computational overheads.

Let us take an example to show this trade-off in more detail. Depending on whether we allow or not the reservation range to vary when handoff occurs, there are two alternatives: static or dynamic. The reservation parameters are fixed while a call is in progress (static reservation range), or they may be updated whenever handoff occurs (dynamic reservation range). The latter provides better mobile-QoS and bandwidth utilization, but requires more overheads.

Bearing these facts in mind, we describe our design principles for bandwidth reservation. The basic idea is to support the mobile-QoS with the emphasis on the simplicity (minimum signaling and computational overhead). First, the bandwidth reservation is synchronized on the time slots same as in [4]. This will reduce the computational overhead significantly compared to the case with no time slots. The size of time slot must be carefully chosen not to degrade the overall network performance. Secondly, the estimation of user mobility is performed only by the measurement. Rather than relying on computation, user mobility is estimated by the aggregated behavior of mobiles that have arrived at each cell. Thirdly, the decision process of reservation parameters is performed in a distributed way. This will keep the required signaling among cells at minimum level. Finally, bandwidth reservation is done only once per call when it is generated. This will also reduce both the signaling and computational overheads even though the mobile-QoS can be worse to some extent.

2.3 Reservation Model

Now we describe the basic reservation model for the bandwidth management scheme to be based on. Figure 1 shows a typical architecture of mobile networks. As shown in Figure 1, each cell can be modeled by the hexagonal geometry structure. Centered on a particular cell (cell A), neighbor cells belong to one of the rings depending on the distance from the center cell. That is, ring- i indicates a set of neighbor cells that are i -hop away from the center cell. Accordingly, cell B and cell C belong to ring-1 and ring-2, respectively. Ring-0 automatically implies the center cell itself.

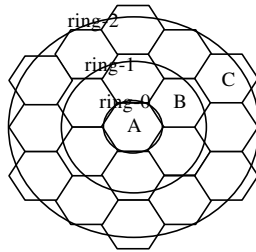


Fig. 1. Cell architecture

Now suppose that two calls requiring bandwidth reservation were generated each from cell B and cell C. Assume also that the reservation range of cell A is at least two

hops and both cells fall within the reservation range of cell A. Figure 2 shows the reservation intervals of these calls that are referred to call B and call C, respectively. From the Figure 2, we need to mention some facts regarding the bandwidth reservation. Comparing the starting times of two reservation intervals, we can conjecture that the estimated speed of the mobile C is slower than the mobile B given that the path lengths traversed by two mobiles are same. It is possible for the call C's starting time is placed ahead of call B's starting time if the mobile C moves much faster than the mobile B. If we assume that two mobiles move at the same speed and traverse different paths, similar arguments can be applied to the estimated path length. We also observe that the amount of bandwidth reservation for call C is less than that of call B. This is due to the fact that cell C is located farther away from cell A than is cell B even though two mobiles request the same level of mobile-QoS. It is also possible that the amount of bandwidth reservation for call C can surpass that of call B if call C requires more stringent mobile-QoS.

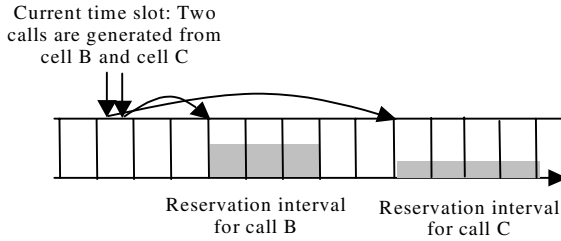


Fig. 2. Structure of reservation slot

3 Bandwidth Management

The proposed bandwidth management scheme consists of two parts: adaptive control of reservation parameters and the associated connection admission control.

3.1 Reservation Control

The reservation parameters mainly rely upon network architecture and user mobility. Given the network architecture, the offered mobile-QoS is dependent on how precise user mobility can be estimated. As stated above, if we want more accurate estimation, it is inevitable to take additional signaling and computational overheads. This may be prohibitive in some situations where fast response time is required.

Our approach to overcome these limitations is to estimate user mobility by a posteriori measurement instead of a priori computation. That is, user mobility is derived from the collection of the arrived mobile's statistics. Then, the reservation parameters are dynamically adjusted based on this measurement-based estimation. By doing this, it is possible to control the reservation parameters adaptively reflecting the current

status of the network. We describe the decision criteria to adjust the reservation parameters in more detail.

Reservation Range. For a particular cell, the frequent arrivals of the mobile-QoS calls with no reservation can be interpreted as the indication that the current reservation range is too small. This can happen when the call holding time is larger or the required QoS is more stringent than estimated. Therefore, if the number of handoff requests by the mobile-QoS calls with no reservation at the cell increases beyond a certain level, we need to increase the corresponding reservation range. On the other hand, if the number of mobile-QoS calls with reservation that do not arrive at the cell increases beyond a certain level, we need to decrease the current reservation range. To do this, we require that each cell maintain the list of mobiles that has reserved bandwidth. For a practical purpose, this list can be maintained within the limited range of time slots (reservation window).

If we are able to identify the origin cell of each handoff call, it is possible to check each neighbor cell separately whether it belongs to the reservation range. We call this non-uniform reservation range since only a portion of neighbor cells in the same ring may join the reservation range. For the simplicity, the reservation range may contain every neighbor cell in the same ring, which is called uniform reservation range.

Reservation Interval. Note that the reservation interval consists of the starting time and the ending time of a mobile with reservation. The starting time must be able to move forward or backward depending on whether the mobile-QoS calls arrive earlier or later than reserved,. On the other hand, the ending time represents the estimation of cell residence time and is dependent on the departure time of the mobile-QoS calls leaving the cell. Similar to the starting time, the ending time also moves back and forth depending on whether the mobile-QoS calls leave the current cell earlier or later than expected.

Reservation Bandwidth. Even though a mobile-QoS call arrives on time at the reserved cell, the handoff request can be blocked when it finds no available bandwidth. It is apparent to increase the amount of reservation bandwidth to prevent this type of bandwidth inefficiency. That is, if handoff blockings for the reserved calls happen too frequently, the amount of bandwidth per reservation must be increased. On the contrary, if the reserved bandwidth is too underutilized, we should reduce the amount of reservation bandwidth and leave more rooms for the newly generated calls.

We describe the proposed management scheme in the form of pseudo code.

// Reservation Range:

```
if (NR1 > TR1)
  if ( (NR2 / NR1 > TRH) and ( RR < RRmax ) )
    { RR++; NR1=0; NR2=0; }
  else if ( (NR2 / NR1 < TRL) and ( RR > RRmin ) )
    { RR--; NR1=0; NR2=0 }
```

Where,

NR1 : Number of QoS handoff request
 NR2 : Number of QoS handoff request with no reservation
 TR1 : Decision value to perform range adjustment
 TRH : Comparison threshold for increment
 TRL : Comparison threshold for decrement
 RR : Reservation range
 RRmax : Maximum reservation range
 RRmin : Minimum reservation range

// Reservation Bandwidth:

```

if (NB1 > TB1)
  if ( (NB2 / NB1 < TBL) and ( BR < BRmax) )
    { BR++; NB1=0; NB2=0; }
  else if ( (NB2 / NB1 < TBH) and (BR < BRmin) )
    { BR--; NB1=0; NB2=0; }

```

Where,

```

NB1 : Number of QoS handoff request
NB2 : Number of accepted QoS handoff request
TB1 : Decision value to perform bandwidth adjustment
TBH : Comparison threshold for increment
TBL : Comparison threshold for decrement
BR : Amount of the bandwidth reservation
BRmax : Maximum amount of the reservation bandwidth
BRmin : Minimum amount of the reservation bandwidth

```

// Reservation Interval (Starting Time):

```

if (NIS > TIS1)
  if ( (NIE / NIS > TIS2) and ( VIS > TImin ) )
    { VIS--; NIE=0; NIS=0; }
  else if ( (NIL / NIS > TIS2) and ( VIS > TImax ) )
    { VIS++; NIL=0; NIS=0; }

```

Where,

```

NIS : Number of QoS handoff request
TIS1: Decision value to perform slot interval adjustment
TIS2: Comparison threshold for increment or decrement
NIE : Number of early arrived QoS handoff calls
NIL : Number of late arrived QoS handoff calls
VIS : Reservation interval (i.e., starting time)
TImax : Maximum reservation interval
TImin : Minimum reservation interval

```

// Reservation Interval (Ending Time):

```

if (NSO > TIS2)
  if (VSO / NSO > AR)
    { AR++; NSO=0; VSO=0; }
  else if (VSO / NSO < AR) { AR--; NSO=0; VSO=0; }

```

Where,

```

NSO : Number of handoff QoS calls
VSO : Residence time of the handoff QoS calls
TIS2: Decision value to perform res. interval adjustment
AR : Reservation interval (i.e., ending time)

```

3.2 Connection Admission Control (CAC)

For the bandwidth management to perform properly, control of reservation parameters is accompanied by CAC. As long as the mobile-QoS is concerned, the CAC is applied separately to new calls and handoff calls. From the viewpoint of mobile-QoS, each call can be divided into two classes depending on whether it requires the mobile-QoS or not: mobile-QoS calls and non-mobile-QoS calls.

Figure 3 and 4 describes the CAC algorithm for handoff calls and new calls using pseudo codes. Here, for the sake of simplicity, we assume that the reservation range is uniform. Let $BW_{req}(i)$, $BW_{res}(i)$, and $BW_{avl}(i)$ denote the requested bandwidth, the reserved bandwidth, and the available bandwidth of a cell that is located i hops

away from the cell where new calls were generated, respectively. In particular, note that $i=0$ indicates the cell where the call was generated (for the case of new call) or the call arrives (for the case of handoff call). Recall that the bandwidth reservation is done only once per call when it is generated and therefore the bandwidth reservation for the neighbor cell is restricted to new calls only.

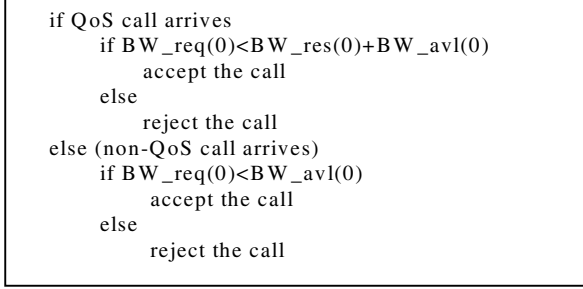


Fig. 3. CAC algorithm (handoff call)

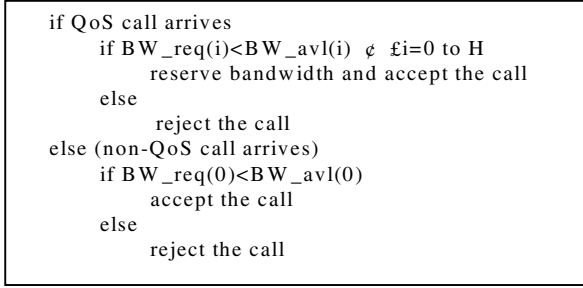


Fig. 4. CAC algorithm (newcall call)

4 Simulation Results

We evaluate the performance of the proposed bandwidth management scheme through computer simulations.

For the simplicity, our assumptions to be used in the simulations are as follows.

- The cell structure is linear (e.g. highway).
- The radius of each cell is uniformly distributed with the average of 1 km.
- The number of available channels in each cell is limited to $C=60$
- The maximum bandwidth is $B=1.6Cb$ where b is the basic unit of bandwidth
- The amounts of reservation bandwidth for mobile-QoS calls and non-mobile-QoS calls are $2b$ and b , respectively
- The ratio of generation rates for mobile-QoS and non-mobile-QoS calls is 3:7.
- The call holding time is exponentially distributed with the average of 180 sec.

- The mobile's speed is uniformly distributed over the range of 0 and 100 [Km/h].

Initial values of the reservation parameters are as follows.

- The initial reservation range is 1 hop and upper bounded by 3 hops.
- The initial reservation interval is 70 sec.
- The initial reservation bandwidth for mobile-QoS calls amount to $b/8$.

During the simulations, the reservation parameters show the following statistics..

- The average reservation range converges at 2 hops.
- The average reservation interval approaches to 120 sec.
- The average reservation bandwidth for mobile-QoS calls increases by 10%.

Throughout the simulations, our main concern is to measure the degree of mobile-QoS focused on the blocking probabilities for new calls and handoff calls. In terms of those parameters, the proposed scheme is compared to the case without reservation. Bandwidth utilization is also compared for two cases.

Figure 5 through 7 show the simulation results. From Figure 5 and 6, we see that the proposed scheme notably reduces the handoff call blocking probability with a slight degradation of new call blocking probability. This degradation becomes almost indistinguishable as the call arrival rate increases. In the Figure 7, we also observe that the proposed scheme shows lower bandwidth utilization. The reason is that the rest of available bandwidth is occupied by reservation instead of new calls.

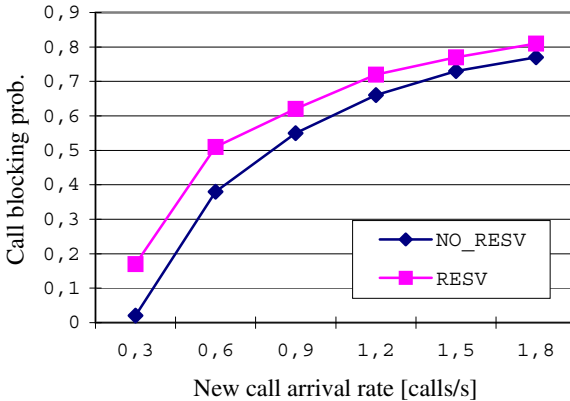


Fig. 5. New call blocking probability

5 Conclusion

In this paper, we proposed a bandwidth management scheme to support the seamless mobile-QoS in mobile networks. The proposed scheme is based on time-selective bandwidth reservation, and can dynamically adjust the reservation parameters depending on the measured traffic conditions. Reservation control was designed as

simple as possible to reduce signaling and computational overheads. However, efforts have also been done not to lose the dynamic features of the proposed scheme.

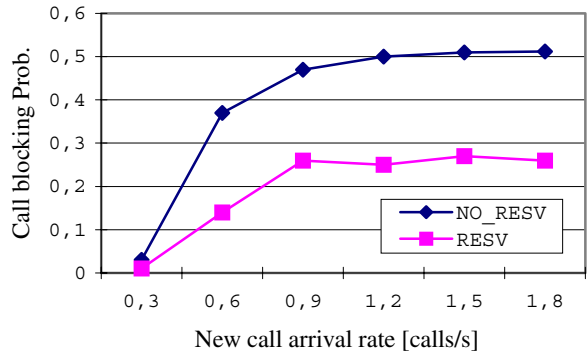


Fig. 6. Handoff call blocking probability

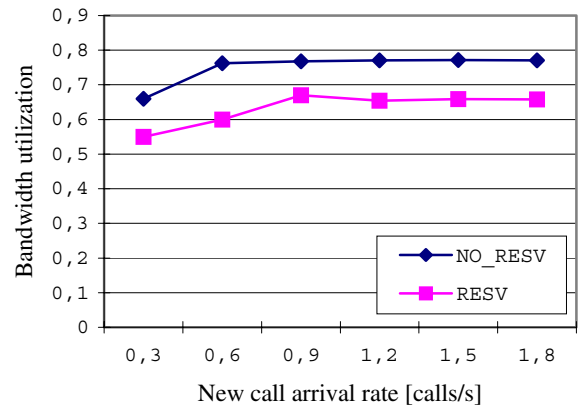


Fig. 7. Bandwidth utilization

From the simulation, we could confirm that the bandwidth reservation provides an effective way to support the mobile-QoS since it could lower the handoff blocking probability as much as you want. However, the bandwidth reservation may easily lead to underutilization and need to be carefully controlled. More works still need to be done to find out optimal reservation parameters in a variety of network environments.

References

1. 3GPP, "Technical Specification Group Services and System Aspects; QoS Concepts and Architecture", 3G TS 23.107, Dec. 1999.

2. A. Acampora and M. Naghshineh, "An architecture and methodology for mobile-executed handoff in cellular ATM networks", IEEE JSAC, 12(8): 1365-1375, Oct. 1994.
3. C. Oliveira, J. Kim, and T. Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks", IEEE JSAC, 16(6): 858-873, Aug. 1998.
4. D. Levine, I. Akyildiz, and M. Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concepts", IEEE JSAC, 5(1): 1-12, Feb. 1997.
5. A. Aljadhari and T. Znati, "A framework for call admission control and QoS support in wireless environment", IEEE Infocom '99.

3G and Beyond & Enabled Adaptive Mobile Multimedia Communication

Theo Kanter¹, Tom Rindborg², and Dan Sahlin²

¹ Ericsson Radio Systems AB, SE-16480 Stockholm, Sweden
theo.kanter@era.ericsson.se,

² Ericsson Utvecklings AB; SE-12625 Stockholm; Sweden
{tom.rindborg, dan.sahlin}@uab.ericsson.se

Abstract. This paper lists various limitations in 2.5G, 3G cellular networks regarding services, and contrasts this to a characterization of fourth-generation wireless networks (4G). In order to investigate 4G's feasibility, cost-effectiveness, necessary functionality, and its potential with respect to services we created a testbed by adding wireless extensions to a public Gigabit IP-network, which was further enhanced with VoIP/SIP to deliver interoperability with existing services and to enable mobile multimedia applications. We present our experiences and analyze the cost-effectiveness of providing (wireless) access to existing voice services in this testbed, how and where these services and end-users can be hosted, and how interworking with services over other networks can be arranged. Furthermore, we present a service architecture to negotiate *adaptive* mobile multimedia communication, with minimal shared service knowledge, which enables applications to adapt to and make optimal use of the heterogeneous mobile infrastructure. We demonstrate the latter by presenting results from building a mobile-aware media-player, and extend it further to take into account the user's context. In conclusion, we show the feasibility and cost-effectiveness of building 4G networks and provisioning of adaptive mobile multimedia applications by extending the emerging broadband infrastructure with existing wireless LANs.

1 Introduction

Previous work [1] has shown that multimedia services, including voice, can be delivered over wireless links with end-to-end IP-connectivity, in addition to the data services that the Internet already provides. This means that the services can be agnostic about the network link layer provided that minimal conditions for delivering the service are met (e.g., latency, bandwidth, upper-boundary for packet-loss, etc.). As a consequence, a third-party application provider can deliver an application to an end-user by *any* network that meets the minimal requirements of the application.

Concerning wireless networks, we need to move the point of integration of these services *out of* the cellular access network in order to enable mobile users to directly interact with Internet content. GPRS (General Packet Radio Service) provides direct Internet access to mobile users and enables the development of multimedia applications for the mobile device. These applications can directly integrate content that resides *anywhere* on the Internet. EDGE (Enhanced Data-rate for GSM

Evolution), the successor of GPRS, will further increase the bit-rate and thereby further relax the limits on the mobile applications and their use of Internet content, thus bringing even more multimedia applications to mobile devices. However, while upgrading *existing* GSM-systems with packet-data services is a logical step with a plausible business case, we should question any steps beyond that from the perspective that alternative technologies and infrastructures are already available and being deployed to provide wireless packet-data services.

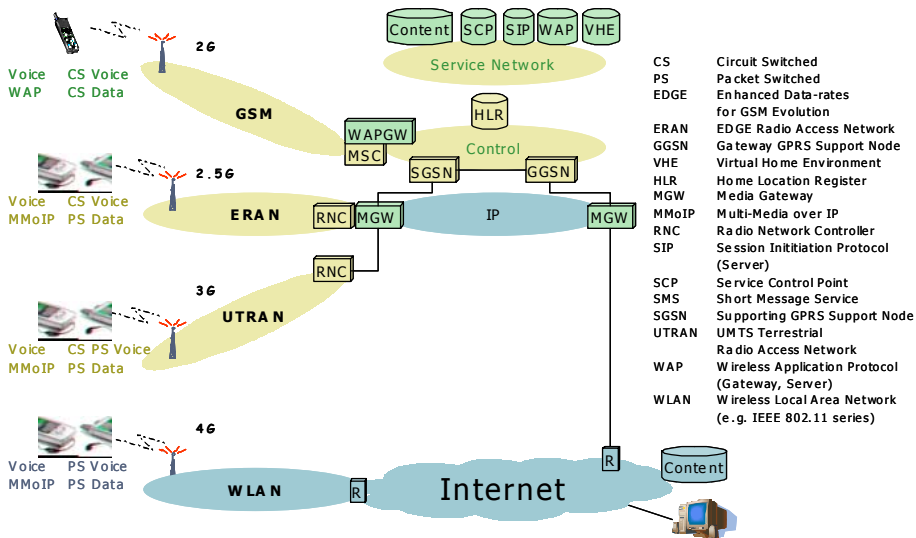


Fig. 1 3G Evolution and beyond

2 Services Architectures (Towards 3G)

This section characterizes the service architectures that are used in 2G, 2.5G, 3G, and beyond. In addition it characterizes the properties of 4G wireless networks. Fig. 1 provides an overview.

2.1 2G

In 2G, mobile devices authenticate themselves and the identity of the user while reporting their location to the Home Location Register (HLR). Speech or data sessions are based on circuit switching of radio channels. A very limited packet data service is provided by SMS. Except for SMS, all services are mutually exclusive. Additional client software in the mobile device (e.g., for Personal Information Management) may be used to invoke the services resulting in so-called Smart-Phones. WAP-clients in the mobile device offer a simple interface to Internet content that can only be accessed through a WAP Gateway, which translates between IP and WAP protocols. A web server on Internet can eliminate the need for an HTML filter by

publishing pages with Wireless Markup Language (WML) tags. Through the use of WML Script content, other services can be invoked (e.g., sending short messages, invoking calls). By following a specific URL, the user can download and play a video from a media server. Using WAP in the mobile terminal causes user services to be strictly dependent on the functionality of the WAP-GW, and thus dependent on the network operator. In addition, circuit switched network access disallows asynchronous application events, this greatly limits the type of services that can be offered to users in a meaningful way.

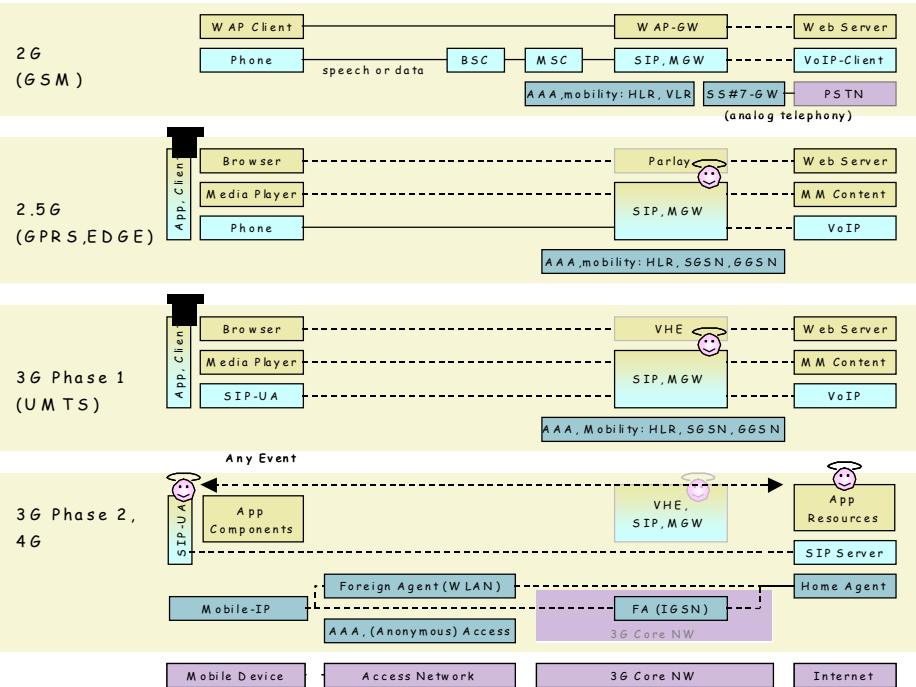


Fig. 2 Service Architectures

2.2 2.5 G

GPRS and EDGE will remove some of these limitations, by offering packet data service. Mobile terminals authenticate themselves to the GGSN and report the location of the user to the HLR through the SGSN. The Mobile Device obtains an IP-address from the GGSN. There are different traffic classes allowing for combinations of switched GSM and packet-switched GPRS traffic. The current standard for GPRS data traffic incurs considerable latency by interleaving data (in order to increase the reliability of data transfer) and to allow for per packet establishment of radio bearers (in order to optimize utilization of radio resources). The operator is still in the position to encourage, if not require that the mobile device be configured to use servers in the operator's service network to setup multimedia sessions.

A SIP server can be used to setup multimedia communication between end-points. This can be further enhanced by adding a Parlay-API [18] to the SIP server in order to execute servlets via a Corba interface on a web server. A web browser can be used for customer control of the services — in what can be regarded as a Virtual Home Environment (VHE), with integrated interfaces to a Service Control Point (SCP), in order to be able to control legacy services. Scripted mobile code can be sent to and executed by agents that are co-located with an application client in the mobile device [19]. Moving the execution of code to the mobile device has various advantages, e.g. performance, and allows the device to report local states back to the server.

Parlay — The Parlay Architecture is based on Corba interfaces that enable hosting of applications outside of specific networks while accessing resources in other networks, through gateways that are installed by the network operator, making these applications and services available to the user irrespective of what network the user is located in. The Parlay API specifications are open and technology-independent, so that anyone can develop and offer advanced telecommunication services.

Clearly we can move services between different networks but only within Parlay domains, but this process is entirely controlled by the network operators. Fig. 3 shows an example of how a simple service using these interfaces can be built. This example was used to prototype wake-up calls and location-dependent information push services in a mobile network.

What is particularly important about this example is that the controlling web interface and the application are only synchronized through network-based servers across a network boundary. Mobile code can be sent to the device to enhance user interaction, but the process must be carried out under the supervision of the application servers and require synchronization across network boundaries. Furthermore, the Parlay interface must be changed each time to reflect capabilities that are present or introduced in SIP [1]. Parlay has these two limiting properties in common with other network-centric service architectures, such as WAP, VHE (see 0), or TINA-C [20].

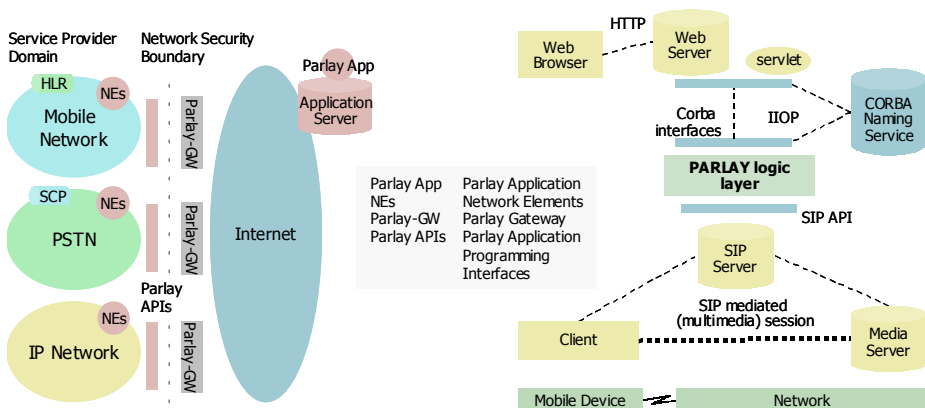


Fig. 3 Parlay Architecture and Prototype of Parlay Access to SIP resources

2.3 3G Phase 1

Mobile terminals authenticate themselves and report the location of the terminal to the HLR through the combined SGSN and GGSN (which also assigns an IP address to the mobile terminal). 3G Phase 1 supports real-time and isochronous multimedia (e.g., voice calls) using end-to-end connectivity over wireless links, which is set up using servers in the operator's service network, to negotiate session parameters regarding quality of service levels (QoS). A Virtual Home Environment ensures that user access to services is independent of the location of the terminal, and that the user interface is independent of the terminal, for instance using (as in 2.5G) a web interface (HTTP) and Java for customer control.

In summary, the service architecture does not differ in principle from the one in 2.5G, and the service architecture offered by an operator of a GPRS, EDGE, or 3G Phase1 network requires any communication, beyond simple browsing of web pages to be *mediated by servers in the operator's Service Network*. Negotiation of services and levels of QoS linked to network specific AAA and mobility mechanisms effectively blocks any possibilities to import or export services to/from Internet ad-hoc. While this service architecture makes perfect sense from an operator's point of view (and follows an established business model), it disallows or in the best case makes it extremely complicated to support the types of communication that we propose. Service mobility between this and other networks can in principle be solved on a per service basis, with adaptations to deal with the specific requirements for mediating functionality in the Service Network. However, we believe this makes services harder to deploy rather than easier!

Virtual Home Environment (VHE) — is a concept for providing personalized service portability across network boundaries and between terminals. The concept of the VHE is such that users are consistently presented with the same personalized features, User Interface personalization, and services in whatever network and whatever terminal (within the capabilities of the terminal) and wherever the user may be located. For UMTS phase 1, VHE consists of GSM services & roaming principles and Service capabilities — see Fig. 4.

The service capabilities offered are call control, location & positioning, PLMN information & notifications, and bearer establishment. It is clear that any services created in this platform are unique to this platform and network (e.g. CAMEL, MExE, SAT — see Fig. 4), and cannot be moved outside of a 3G network to the Internet, and this architecture is not open to executing random services on the Internet.

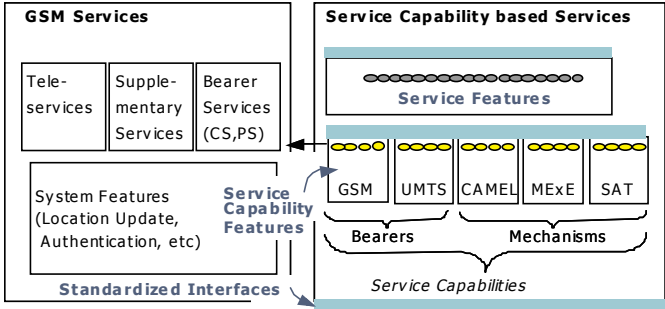


Fig. 4 VHE Realization

2.4 3G Phase 2

In 3G Phase 2, Mobile IP is used for handoffs and roaming between 3G and other networks; hence user mobility is no longer controlled by the (3G) GSN nodes. Mobile terminals authenticate themselves to AAA-servers via the integrated SGSN and GGSN node (IGSN), which also acts as a foreign agent for mobile-IP, thus assigning an visiting IP address to the mobile terminal. The IGSN reports the location change to the home agent of the mobile terminal and forwards the AAA information to the HLR for charging purposes. This AAA and mobility scenario enables the mobile to negotiate communication with resources outside of the 3G networks without intervention of servers in the operator's Service Network. Naturally, the operator can offer support for different levels of QoS and even differentiated charges, but the fact remains that the services are negotiated end-to-end, and not simply inside the operator's Service Network. However, mobile terminals are required to have detailed knowledge of such support services, which may differ between networks and may change over time. Thus, we need a means to describe shared knowledge of support services and also means to automatically obtain such knowledge in order for services and mobile devices to migrate between networks. This is one of the design goals of the extensible Service Protocol [16], see also section 0.

3 Fourth Generation Wireless (4G)

Since multimedia can be delivered with end-to-end IP connectivity over wireless links [1], this allows us to extend all existing voice services to these networks. So-called 'hot spots' equipped with wireless LAN (WLAN) extensions to the Internet are becoming available, and today provide us with even higher bandwidths (e.g. 11 Mbps in IEEE 802.11b), for example Telia's HomeRun [3] system, corporate WLANs, and "semipublic" WLANs.

This is particularly important, since broadband Internet access is being provided in a rapidly increasing number of public locations (hot spots) and even homes in urban areas. The provisioning of broadband Internet access is being installed / provisioned by power companies, transportation companies, housing co-operatives, joint-ventures of municipalities, etc., all of whom have a radically different business model than traditional telecom vendors and operators of cellular networks. Extending this packet-switched infrastructure with wireless access points, such as IEEE 802.11b Wireless LAN is straightforward. In addition, mobility solutions, such as Mobile-IP, and IPv6 are available to provide the necessary scalability that accommodating millions of users and devices will require [4,5].

Furthermore, solutions for direct access to Internet (not requiring an existing subscription, but rather a direct settlement, e.g. with E-cash) are available [6]. In fact, such operators simply provide IP-access, they *do not* necessarily even need to do authentication, authorization, and accounting (AAA), since they get paid directly or indirectly.

Consequently, users with mobile devices can, in principle, use any service from any third party, without any intervention by the operator that provides the network access. It should be noted that attempts to limit the customer's choice by incumbent operators have been found to violate the EU's competition laws.

Thus, the properties of 4G are such that it provides users with (1) multimedia over end-to-end IP (wireless) links with (2) high-bandwidth, between (3) multiple, heterogeneous, access networks, and with (4) direct access to the Internet and thus end-to-end IP-connectivity to (5) third-party mobile multimedia services, without the need for prior subscription for Internet access with these access network operators.

4 Problem Statement

Two novel aspects characterize the resulting fourth generation wireless network scenario (4G). First, the network consists of a conglomeration of heterogeneous networks that provide end-to-end IP connectivity over wireless links. In addition, aside from mobility support for devices (e.g., Mobile-IP) and support for direct anonymous public access, it is a "stupid network" scenario, where the network only provides packet transport, and therefore it is an "operator-less" network with respect to services.

The network scenario for 4G networks as outlined in the introduction appears to be straightforward, but we need empirical results in order to know how easy and cost-effective it really is to provide wireless access to the Internet to end-users. In addition, this network has to provide access to existing services of which telephony (voice) is the most important one.

We are also in the position to provide *new* mobile multimedia applications that take advantage of the fact that end-points have computing capabilities, and through sensory capabilities can have knowledge about the context of users. Furthermore, the "operator-less" service scenario also implies that the mobile users and devices that participate in communication over 4G must become smarter, i.e., they must be able to respond to a wide range of events:

1. Other users, mobile devices, and communication resources may become "visible" in an ad-hoc fashion, either by proximity, or actively communicating.
2. Entities (users, mobile artifacts, and virtual objects) may exchange events that range from simple invitations to join a session all the way to manipulations of shared virtual objects.
3. The communication conditions will vary between and *even within* access networks. This is especially the case where wireless communication is concerned. Applications must be able to act reasonably given knowledge of the situation.

An application architecture for such adaptive, mobile personal communication has been described in [7,16] featuring mobile agents, VoIP/SIP-enabling multimedia applications, using end-to-end IP communication between users over wireless links.

The questions are thus how to provide easy public access to these services to end-users, understand what functionality is needed, and to be able to demonstrate the feasibility, cost/effectiveness of the architecture, and its potential regarding applications. Our hypothesis is then that such an approach to 4G is both practical and feasible, and that it can be enabled to bring such adaptive mobile multimedia applications to end-users. These were our purposes for building an experimental fourth-generation wireless test bed infrastructure, and verifying our application architecture by prototyping, and our results are shown below.

5 Experimental Network

We have built an experimental fourth-generation wireless testbed by extending Internet42, an existing Gigabit-Ethernet IP-network [9] (Fig. 5).

The project involved several parties: Ericsson Radio, Royal Institute of Technology (KTH), Telia, and Brf Bågen. Besides points of presence at research facilities (Ericsson Radio, KTH, and Telia) in the Stockholm suburbs of Kista, Älvsjö, and Farsta; Internet42 also has a point of presence in the center of Stockholm where it provides, at low cost, 100 Mbps network Internet access to each apartment in a large housing co-operative, Brf Bågen [10]. We have extended the services of Internet42 [9], by adding 11 Mbps wireless packet data access points (IEEE 802.11b), agent servers, media servers and content management, voice gateways (VGW) with anonymous direct access to Internet (DIA), support for device mobility (Mobile-IP) and service mobility (SIP). We will add GPRS early next year to our test bed. The functional components are further explained starting in the sections below.

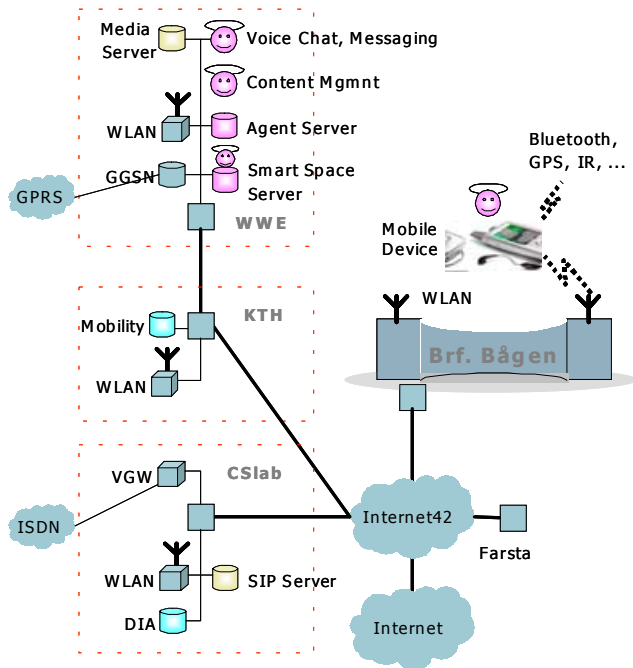


Fig. 5 Experimental Network Overview

5.1 Brf Bågen

In April 1998 the housing cooperative Brf Bågen in central Stockholm installed a local area network in all 261 apartments and in all companies located in the buildings [10]. The main use of the LAN is to provide Internet access through a leased 2Mb/s line.

Gigabit Ethernet is used both to connect the buildings to the Internet42 backbone and between the five Ethernet switches, which provide each user with 100 Mb/s Ethernet to the Ethernet switch. The housing cooperative acts as an operator with the following distinguishing characteristics: (1) Users get real IP-numbers, either statically (for servers) or dynamically through DHCP, (2) there is no firewall to the

Internet, and (3) there is no restriction on traffic, neither between the users nor to the Internet.

The only local services provided are mail, local personal web pages, and local news. Currently 56% of the apartment owners are actively connected to the net. A few companies are also connected, and share the bandwidth with the apartment owners. The residential LAN infrastructure has worked very well with the exception of a few prolonged interruptions on the Internet connection, which led Bågen to change ISP after a completely open tender. In important note is that this possible and relatively easy due to the fact that Bågen *owns* the LAN.

5.2 Wireless Access

Extending a fixed ethernet network with wireless-LAN access points (IEEE 802.11b) near the points of presence in research facilities is straightforward. Adding wireless LAN to a housing co-operative and thereby providing wireless access to Internet over its infrastructure in a public space is a different matter confronting the housing co-operative with both technical issues (mainly security) and non-technical questions (concern about



Fig. 6. IEEE 802.11b Coverage at Brf. Bågen in Stockholm (Photograph Courtesy of Lantmäteriet © 2000)

antenna aesthetics). [3,6,14]. The effort and cost to provide broadband wireless packet data was very low. As usage grows we can add access points. With a single access point we obtained good coverage in a large public space at low cost (Fig. 6), as the cost of hardware was \$2600, and the area covered 200m in radius = 126000 m², thus equal to \$0.02/m². Users share up to 11 Mbps of bandwidth via a single access point, but wireless LAN technology (802.11b) allows us to add access points as the user density and demands increase. Monitoring throughput during videoconferences we observed 80-90% network utilization.

5.3 Direct Internet Access

Wired Equivalent Privacy (WEP) in IEEE 802.11b has a dual purpose of authenticating users and providing data encryption with the following disadvantages: (1) WEP differs between manufacturers, (2) WEP encryption keys must be manually distributed, (3) WEP is set on a per-network basis rather than on a per-user basis, and (4) Windows-based machines must be rebooted after a key change. Alternatively, the WLAN infrastructure can be complemented with an authentication mechanism based on pre-shared or certificate-based keys. However, this approach precludes anonymous roaming access. Therefore we used a third method Direct Internet Access (DIA) [6],

which provides anonymous authentication and allows the access provider to charge via eCash. This approach thus makes access authentication keys redundant, and allows simple roaming access. Consequently, there is no reason to do accounting or administration of users. Additional security is ensured using IPsec and IKE [14].

5.4 SIP

A SIP redirect server allows end-users to register with a SIP URL and enables others to send them invitations to multimedia communication (enabling personal service mobility). Thus, assigning these identities to Personal Agents allowed us to leverage its functionality to easily implement remote customer control of personal messaging (via web pages) and Internet Telephony (e.g. diverting calls when in a meeting), where the voice gateway allows us to locate agents locally or remote as SIP URLs by identifying telephone numbers and vice versa. When we allowed the personal agent to monitor incoming calls to its number via the telephony-GW using a group number, then a consistent and complete (i.e., messaging, Internet-, and switched- telephony) solution was achieved for personal communication. SIP invitations can now also be sent through firewalls and with NAT [11], which might allow a local network access operator (e.g. Brf Bågen) to increase its security while preserving all services.

5.5 Mobility

Strategies using Mobile-IP or other network mobility protocols can be used to support handoffs [4,5]. We used the Mosquito Net Mobile-IP stack [19] to enable our devices to do handoffs between GSM-data and WLAN, in order to investigate the feasibility to do VoIP handoffs. These attempts proved to be unsuccessful due to various reasons: GSM-data session setup times through a dial-in connection are too time consuming, which is fixed by using GPRS. However, we found that infrequent agent advertisements (minimally 1 sec. Delays - RFC 2002) overshadow the 500 msec latency in the GPRS air interface. Thus a modified approach is needed (e.g., with micro-mobility). A separation is needed between mobility for voice and the mobile device, so as to circumvent unnecessary delays due to triangular routing, where location changes are used to send voice packets to the new address, either via SIP or simply using RTP, thus resulting in minimal delays.

5.6 Quality of Service

As there was ample bandwidth, the use of speech codecs was unnecessary from a QoS perspective. There was no perceivable packet-loss from the perspective of the end-user. However, in areas where the signal is weak, we may benefit from using robust header compression [12].

When end-to-end security needs to be guaranteed, then IPsec is an obvious choice. When speech and signaling use different ports then the increased header length can be dealt with separately by applying ROCCO [15] to this stream of IP-packets without requiring a trust relation to be established between the mobile device and the access

point. This allows us to establish the direct Internet access strategy without AAA-functionality, lest it be necessary for commercial or operational reasons.

5.7 Capacity

In addition, capacity and spectrum efficiency will benefit from using robust header compression [12], as IP-headers incur a considerable overhead with respect to the size of content for this streaming audio. Furthermore, speech compression will further up the number of simultaneous voice users. In ideal cases, with robust header compression, and compressing 16 Kbps PCM to around 6 Kbps G.723, the number of possible simultaneous voice users, in a single 11 Mbps cell could be well over one thousand, corresponding to 10^5 users in a macro cell, with the possibility of adding more access points if more capacity is needed.

5.8 Hosting, Interworking

An important aspect for parties such as Brf Bågen whose focus it is to make connectivity available in the infrastructure but are not interested in directly operating services, is that should be able to outsource hosting of substantial parts of the functionality. This is supported by our architecture, where it is of no concern where the components are located as long as they are available on the Internet. All this functionality could even be packaged as a do-it-yourself 4G kit, since management of the functionality is not necessarily more complex than maintaining a web site.

Interworking with other Internet access providers, both wireless (e.g., Telia Homerun [3]) and fixed networks needs to be addressed at two levels. First, the user must be allowed access to and roam between networks. There are technical solutions available for both. Second, there must be an agreement between parties who decide to allow roaming between others networks. This can be supported by clearinghouse, thereby relieving parties of managing mutual agreements.

6 Enabling Adaptive Mobile Multimedia

6.1 eXtensible Service Protocol

Proprietary protocols between the agents would quickly add up to unmanageable complexity in the system. A generic protocol needed for learning and conveying the capabilities to communicate with a resource is addressed by the eXtensible Service Protocol (XSP) [16], an XML-based protocol, which allows agents to communicate capabilities to other agents, in order for them to use each other's methods. In our prototype we use an XSP-enabled agent to act as Mobile Interactive Space, with which other agents can register, subscribe to events, query for other entities' presence, properties and methods. They can invoke each other's methods, set properties or extend each other's capabilities, using XSP.

A personal (mobile) agent running in the mobile device can use SIP for (1) provide user mobility, (2) session invocation of arbitrary resources (e.g., voice, chat, etc.). While SIP delegates session signalling to the invoked resource components (e.g. VoIP-clients), the agent supervises what communication is going on for the user. The agent is able to monitor events, make intelligent decisions about the user's communication context, contact other agents if need be, and invoke communication, and leverage the fact that we can use sensors on the mobile device [13], or in the environment (e.g., GPS) for even more flexible adaptation of the communication to the user's context [7]. We used this model for prototyping user context-dependent information retrieval and voice-communication invocation, presented in the next section.

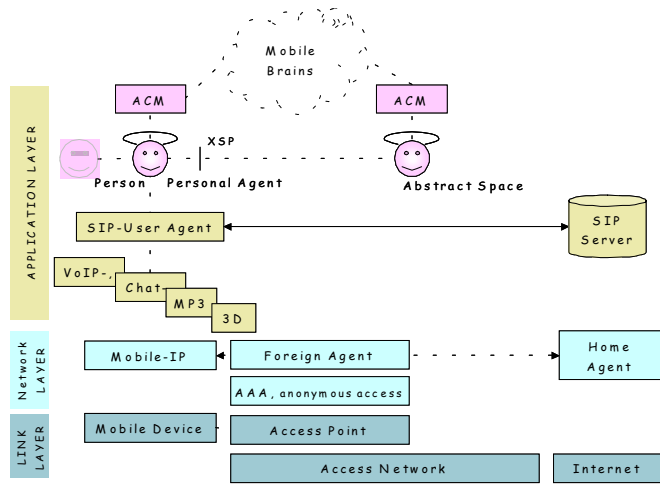


Fig. 7. Application Architecture and Functionality

6.2 Smart Delivery of Multimedia

We have created a Mobile Aware Media Player that takes into account user movements in the network and the resultant changes in communication conditions, as well as on-going negotiation of content delivery according to the availability of (new) multimedia content on Internet media stations and intermediate media stores in the access networks [2]. The Personal (mobile) Agent uses the eXtensible Service Protocol for the necessary flexible negotiation between entities (Internet media stations, intermediate media stores, and end-users) [16].

The Personal (mobile) Agent connects to an Internet Media Station with MP3 content, which in turn diverts communication to a Content Proxy Agent in the access network. The Content Proxy also extends the functionality of the Personal Agent by sending a protocol object for streaming and playing out MP3-audio using RTSP when the user is on-line. Multimedia delivery is redirected to an optimal point of access from a user (price/performance) perspective, based on user context information: e.g., Access Network Agents notify Content Proxies in the access network of available bandwidth, and the Location Agents provide location prediction information, on the basis of which the Content Proxy the Personal Agent decides to receive content in a hot spot with 802.11b WLAN.

In addition, this approach is used to mitigate between bandwidth demands for both switched mobile and remaining bandwidth for this service over GPRS in relation to other services, to the extent that we can utilize of unused frames.

Furthermore, we improved our agents to recognize resource URLs from IR beacons [13,17] to invoke the automatic playout of multimedia content that was associated with this device. Thus we can attach beacons to various locations at Brf Bågen and demonstrate multimedia that is associated with different locations in that area to visiting mobile users.

These results are particularly important because they provide additional support for our hypothesis

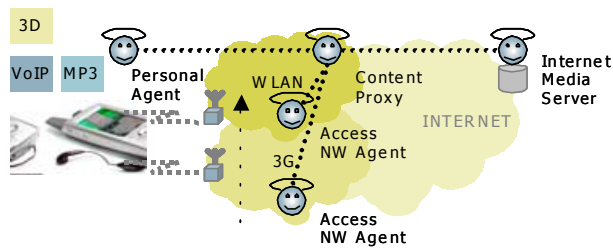


Fig. 8 Smart Delivery of Multimedia

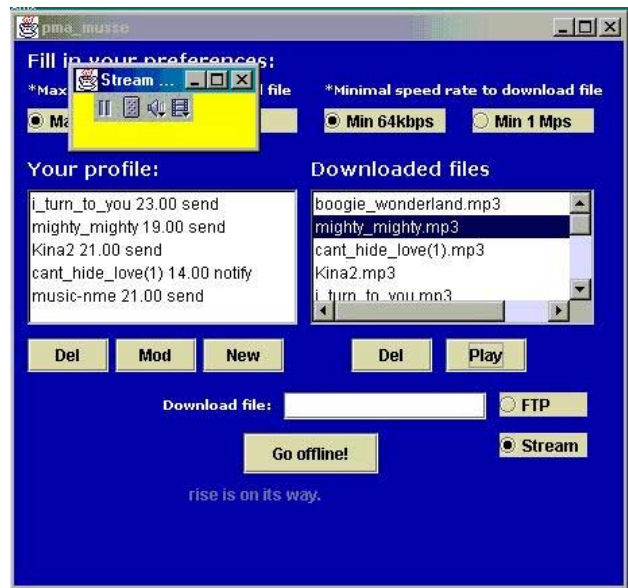


Fig. 8 Streaming Content at 64 Kbps (minimum)

7 Conclusions

Our contribution describes provides real world experiences from pioneering building a large-scale deregulated multimedia enabled mobile Internet, to which existing voice services have been moved successfully, and in which a novel open application software architecture (i.e., mobile agents and a novel extensible service protocol) leveraged the combined mobility and flexibility of end-to-end IP over wireless in entirely new classes of applications at the intersection of mobile and ubiquitous computing and cellular telephony, and we provide evidence of its feasibility or practicality:

Thus, in addition to the properties of 4G in section 0, our approach to 4G offers functionality that can be installed and managed by end-users themselves or such organizations whose business concept is likely to be focused on deriving revenues other than providing Internet access. This functionality must lend itself to be packaged as self-manageable functionality, or be outsourced in case that is a more

suitable model for network operation. The recipe for putting up such a do-it-yourself 4G is to simply:

1. Package functionality (e.g., voice gateway, SIP server, agent server, etc.)
2. Negotiate fixed Internet access (e.g., xDSL, cable, fiber, etc.)
3. Decide whether you want the functionality hosted.
4. If so, just connect the antennas and put them on your roof.
5. Negotiate your service level with the clearinghouse and have it announced.

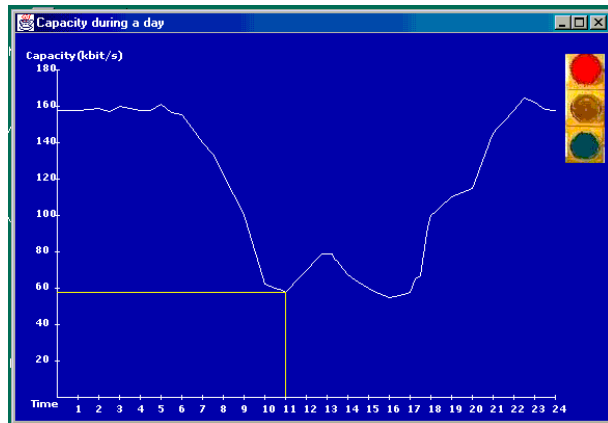


Fig. 9 Redirection due access network threshold

We have provided wireless LAN (IEEE 802.11b 11 Mbps) based public access to the Internet in various public locations in Stockholm, one of which the housing co-operative Brf Bågen. Thus, we have shown that providing functionality to provide end-users with broadband end-to-end IP communication over wireless links using direct access to Internet is perfectly feasible, cost-effective, and enables adaptive mobile multimedia communication. This functionality can either be packaged or outsourced to become usable and manageable by end-users of organizations whose focus is not primarily to own or operate networks. Thereby, we create a public broadband wireless Internet infrastructure that is not regulated in any sense and can, in principle be used by anyone. This unregulated infrastructure provides connectivity to services that, irrespective of whether they are located locally or elsewhere, are not part of the network. Any transactions between end-user and the application service provider, is conducted without the network access provider having any knowledge or role. On the other hand, an application service provider may benefit from being able to get support from the operator for end-users to maximize performance of their service in these hotspots. Such a scenario has been outlined in [1], in which case it is plausible that the provider of a hot spot will be compensated, and thereby creating additional incentive for putting up such networks. Furthermore, we demonstrated a mobile application, which given our application architecture, enables users and mobile devices to negotiate communication that take into account user context and communication conditions.

In conclusion, by virtue of our results, we claim that development of infrastructure for 3G should be aligned according to the criteria that were discussed in this paper. Furthermore, R&D efforts regarding applications should focus on enabling mobile multimedia communication in such a deregulated infrastructure by adopting a service architecture that promotes deregulation on an application level [16].

8 Future Work

In order to fully understand the challenges when deploying unregulated 4G infrastructure along the lines of this paper we will further address providing anonymous access to Internet, including AAA. We have already started an investigation regarding the role and functionality of a clearinghouse between access operators for our scenario.

Furthermore, we will further investigate our network scenario's potential with respect to applications and enabling mechanisms. We have started to prototype novel applications that are based on the application architecture as described in [2], such as context-aware 3D-space that can be shared by mobile users who can have simultaneous voice communication by means of VoIP. Furthermore, results of prototyping the eXtensible Service Protocol [16] for negotiation of ad-hoc application between multiple users and/or devices will be shown during 2001.

References

1. M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg - RFC 2543 on SIP: Session Initiation Protocol, IETF/Network Working Group – March 1999.
2. T. Kanter, P. Lindtorp, C. Olrog, G. Maguire, "Smart Delivery of Multimedia Content for Wireless Applications", to be presented during the 2nd International Workshop on Mobile and Wireless Communications Networks (MWCN'2000), May 2000.
3. Telia HomeRun: <http://www.homerun.telia.com/>
4. E. Wedlund, H. Schulzrinne, "Mobility Support using SIP", Proceedings of the 2nd ACM International Workshop on Wireless Mobile Multimedia (WoWMoM'99)
5. A. Campbell, J. Gomez, S. Kim, C. Wan, A. Valko, "A Cellular IP Testbed Demonstrator", Proceedings of the Sixth IEEE International Workshop on Mobile Multimedia Communications (MOMUC'99) (p. 145-148)
6. P. Jokela, "Wireless Internet Access Using Anonymous Access Methods", Proceedings of the Sixth IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99), November 1999.
7. T. Kanter, "Adaptive Personal Mobile Communication", Licentiate Thesis, KTH, March 2000.
8. T. Kanter, "Adaptive Personal Mobile Communication: An Extendable Protocol and Application Architecture for Mobile Interactive Spaces", PhD Thesis Proposal, KTH/IT, May 2000.
9. Internet42: Public Gigabit-Ethernet IP-network connecting housing co-operative BRF Bågen in Stockholm (<http://www.fatburen.org>)
10. Dagens Nyheter, "Folke surfar på prov i gräddfilen", April 29th 1998.
11. F. Thernelius, "SIP, NAT, and Firewalls", Master Thesis, Royal Institute of Technology (KTH), May 2000.
12. L-E. Jonsson, M. Degermark, H. Hannu, K. Svanbro, "RObust Checksum-based header Compression" (ROCCO), IETF Network Working Group Draft, September 1999.
13. Mark T. Smith, "SmartCards: Integrating for Portable Complexity", IEEE Computer, August 1998, pp. 110-112, and 115.
14. F. Moioli, "Security in Public Wireless LAN Networks", Master Thesis, Royal Institute of Technology, June 2000
15. X. Zhao, M. Baker, "Flexible Connectivity Management for Mobile Hosts", Technical Report: CSL-TR-97-735, September 1997
16. T. Kanter, "An eXtensible Service Protocol for Adaptive Personal Mobile Communication", Submitted to the International Workshop on Smart Appliances and Wearable Computing (IWSAWC'2001) 2001.
17. T. Kindberg et al, "People, Places, Things, "Web Presence for the Real World", Proceedings of the 3d IEEE Workshop on Mobile Computing Systems and Applications, December 2000, Monterey USA.
18. PARLAY Forum: <http://www.parlay.org/>
19. R. Glitho and A. Wang, A Mobile Agent based Service Architecture for Internet Telephony, International Switching Symposium 2000 (ISS00), Birmingham. May 2000.
20. TINA-C Service Architecture Version: 5.0, Date: 18 June 1997

Creation of 3rd Generation Services in the Context of Virtual Home Environment

José Oliveira^{1,2}; Renato Roque³; Simone Sedillot⁴; Eurico Carrapatoso^{5,2}

¹ Universidade do Porto, Faculdade de Economia do Porto (FEP),
Rua Dr. Roberto Frias, s/n 4200-464 Porto, Portugal

² Instituto de Engenharia de Sistemas e Computadores do Porto (INESC Porto),
Pç. da República, 93 r/c, 4050-497 Porto, Portugal
{jmo, emc}@inescporto.pt

³ Portugal Telecom Inovação (PT IN), Lg. Mompilher, 22, 2º, 4050- 392 Porto, Portugal
rroque@ptinovacao.pt

⁴ Institut National de Recherche en Informatique et Automatique (INRIA),
Domaine de Voluceau-Rocquencourt, 78153 Le Chesnay, France
simone.sedillot@inria.fr

⁵ Universidade do Porto, Faculdade de Engenharia da Universidade do Porto (FEUP),
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal

Abstract. Virtual Home Environment is a new concept that emerged in the context of the 3rd generation networks for mobile communications. The objective of this paper is to present the work that currently takes place in VESPER¹, an IST project in the area of the VHE. The paper presents the VESPER project approach to the VHE architecture, its main components, the way they interact and the way this architecture will facilitate the creation of services embedded with the VHE concept. The paper presents also two demonstration services, their main functional features and how they use the VHE architecture defined in VESPER.

1 Introduction – The VHE Concept

Virtual Home Environment is a new concept that emerged in the context of the 3rd generation networks for mobile communications. The 3rd Generation Partnership Project (3GPP) (a standardisation body from the European Telecommunications Standards Institute - ETSI) [1] defines VHE as: “a concept for Personal Service Environment portability across network boundaries and between terminals. The concept of the VHE is such that users are consistently presented with the same

¹ This work has been performed in the framework of the project IST VESPER, which is funded by the European Community. The authors would like to acknowledge the contributions of their colleagues from *Intracom Hellenic Telecommunications and Electronics Industry S.A.*, *National Technical University of Athens*, *Institut National de Recherche en Informatique et Automatique*, *IKV++ GmbH Informations und Kommunikationstechnologie*, *GMD - Forschungszentrum Informationstechnik GmbH*, *Fondazione Ugo Bordoni*, *Universita' Di Catania*, *Portugal Telecom Inovação*, *University of Surrey*, *Technical Research Centre of Finland* and *SIEMENS AG Osterreich*.

personalised features, User Interface customisation and services in whatever network and whatever terminal (within the capabilities of the terminal and the network), wherever the user may be located” [2].

This innovative concept appeared when the terminals and networks capabilities rapidly increased, enabling the emergence of a wide variety of highly sophisticated and personalised services over the widest possible coverage area. In this context the user wants to subscribe services that appear to be the same from the user’s perspective, regardless of the access point in use, using even different types of terminals, and independently of the physical realisation of the service.

Important aspects that should be covered in a virtual home environment service provision context are:

- Personalisation of Service Environment
- Adaptation of Service Environment
- Service Portability
- Service Session Mobility.

Personalisation means the ability of the user to personalise and modify the services that he subscribes, choosing the way to access them (network/terminal preferences) and to decide on the behaviour/aspect of the user interface. The service provider should keep this personalisation, as much as possible, independent of the user terminal and access point.

Service adaptation covers terminal and network adaptation. The service must be adaptable to different kinds of terminals, considering their potentially different capabilities. Network adaptability refers mainly to quality of service (QoS) adjustments (usually downgrades, but also upgrades), which may be necessary if the network conditions change or when the service is offered in different terminals.

A portable service should be accessible from the user’s home network and from various alternative networks with the same look and feel.

Session mobility allows a user that has an active session on a particular terminal to move that session to another terminal (e.g. by suspending it on the first terminal and resuming it on the second terminal).

The objective of this paper is to present the work that currently takes place in VESPER, an IST project in the area of the VHE. The paper presents the VESPER project approach to the VHE architecture, its main components, the way they interact and the way this architecture will facilitate the creation of services embedded with the VHE concept. The paper presents also two demonstration services, their main functional features and how they use the VHE architecture defined in VESPER.

2 VESPER Project

In the scope of the European IST programme, the VESPER project (Virtual Home Environment for Service PErsonalization and Roaming users - IST-1999-10825 (Key Action 1.1.2-4.2.4)) intends to define and develop an architecture for the Virtual Home Environment realisation and validate it with some demonstration services [3]. This architecture must provide ubiquitous service availability, personalised user interfaces, service portability and session mobility, while users are roaming or changing their terminal equipment. The VHE should hide away from the user the

variety of access network types (fixed or wireless), the variety of supporting terminals, and the variety of the involved network and service providers responsible for service provision.

The project adopts an incremental process for the VHE specification, which will be carried out in three phases. The initial two phases will provide output to be tested in demonstrations and the third will benefit from the test feedback in order to produce a whole and justified result.

3 VESPER Architectural Approach

VESPER aims at offering an architectural solution and an implementation of the VHE with the capability of providing services to end users with a consistent look and feel, independently of location, serving network and terminal technology. It also facilitates service adaptation to different network environments supporting directly connected, cordless and cellular access.

VESPER concentrates on VHE aspects not explored before, while at the same time it proposes improvements in other aspects, thus achieving the specification of a complete VHE architecture.

Key innovation areas identified in VESPER are:

- Service continuity;
- Service scalability (i.e. adaptability to network, terminal characteristics);
- Service personalisation (i.e. definition, standardisation and management of service and user profiles).

The VHE architecture, defined by the VESPER Consortium, derives from a qualitative combination of selected architecture concepts, coming from several sources: Telecommunications Information Networking Architecture (TINA) [4], PARLAY [5], Open Service Architecture (OSA) [6,7], Telecommunications Service Access and Subscription (TSAS), from the recent work within standardisation bodies, such as 3GPP and ITU, and from advanced software technology, including distributed processing and agent technologies.

To specify the service architecture enabling the VHE concept, the project identified all the VHE requirements [8]. From that the VESPER project defined a VHE service architecture that is represented in the following two figures.

In Fig. 1 the reference VHE architecture may be seen. On the left side the terminal with the possibility to have VHE support (in terminals where it is possible to install software) is represented and on the right side the VHE Provider (represented by the VHE components), the application server and the different networks may be seen. The transparency between the VHE Provider and the different networks is achieved by the OSA/Parlay gateway.

In Fig. 2 are depicted the set of VHE components identified by VESPER (the VHE components box represented in the network side of Fig. 1). The components can be accessed through the User, Administrator and Service APIs and offer the main VHE features to these three actors.

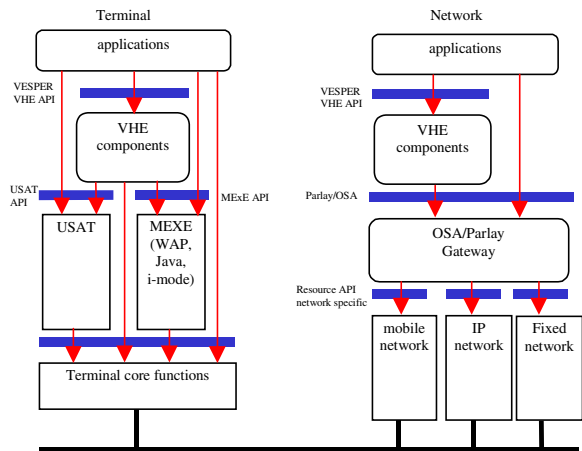


Fig. 1. Reference VHE architecture for VESPER.

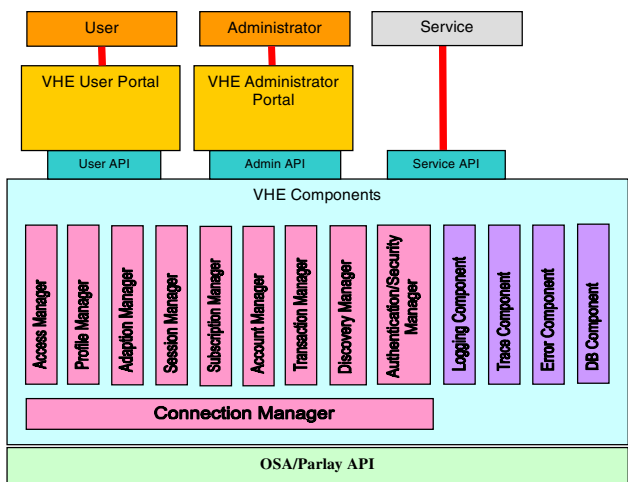


Fig. 2. VHE components.

VESPER defined as well a Roaming Model applicable to the VHE, which comprised a high level definition of the entities, domains and roles involved and all the interactions among them. A Session Model was also defined by the project. Three different types of sessions were identified: **VHE Session** (a temporary association between a user and a VHE Provider to allow him to use VHE or Value Added Service Provider (VASP) services); **VHE Service Session** (a temporary association between parties (peer parties or client and server) through the mediation of the VHE system. The association may subsequently include a VASP Session); **VASP Session** (a temporary association between a number of peer parties or between clients and servers interacting according to a VASP service).

4 Creation of 3rd Generation Services

The emerging 3rd generation networks will enable the existence of very attractive services to the end-users. With the capability of transferring audio, video and data at high rates, a large variety of services, expected to be far more valuable to the user than the current ones, will rapidly appear.

As these services will be available also over mobile networks, a natural requirement from the end-users will be to access personalised services from any place, transparently and independently of the underlying network technology and the access point. But this functionality will not be feasible if there are not ways of facilitating the creation of services, already embedded with those concepts.

VESPER specifies how the architecture components that specifically provide VHE concept features can be associated with the OSA/Parlay frameworks. These frameworks are open interfaces between the network and the service provider, allowing the service provisioning, independently of the underlying network technology.

Open Service Access (OSA) and Parlay (respectively defined in the contexts of 3GPP and Parlay Group) provide frameworks and APIs suitable to create services based in standardised service features offered by different network technologies. This is achieved with an open programming interface (API), which allows applications to access the functionalities of the network and some generic support functions in a secure way. These functionalities include call control services (generic two-party calls, multi-party, conference), user interaction services, messaging services and mobility services (user location and user status).

VHE components will provide as well an open API to VASPs, enabling and facilitating the VHE concept within the service. Features such as personalization, session mobility or adaptation to different terminals offered by VHE providers will be available to services that use this API.

To validate the VESPER architecture, the consortium has decided to develop some applications, namely a Customer Care service and a Calendar service. These applications will use the components provided and in order to carry out their development the approach was to identify common usage scenarios that fully describe the interactions between the users and the VHE Provider (represented here by the VHE components) and between the VHE Provider and a service offered by a VASP.

The sequence diagrams presented next describe examples of these scenarios common to all services provided in the context of the VHE. These “standard” ways of using various VHE aspects will facilitate the creation and provision of services by the service providers.

In the description of the scenarios such as login to VHE Provider, start a service, terminate a service or suspend a VASP session, the VHE specific aspects that bring benefits to the services and to the users are emphasised.

User Login to a VHE Provider and Start a VASP Service

The user login to the VHE Provider (1) and is authenticated by the system (2). If the authentication succeeds, a VHE Session is created (3) and the accounting context of this specific user is set (4). The Access component obtains from the Discovery component the list of services which the user is subscribed to (5).

The VHE Provider enables the user to own several User Profiles (UP) each containing different Service Preferences (SP) for the same service and the same subscriber. Therefore after the user selects a service to use (6), the Access component provides a list of UPs that contains preferences for the service he wants to use (7). The user selects one UP (8) from this list. This choice will affect also the look and feel of the service provision.

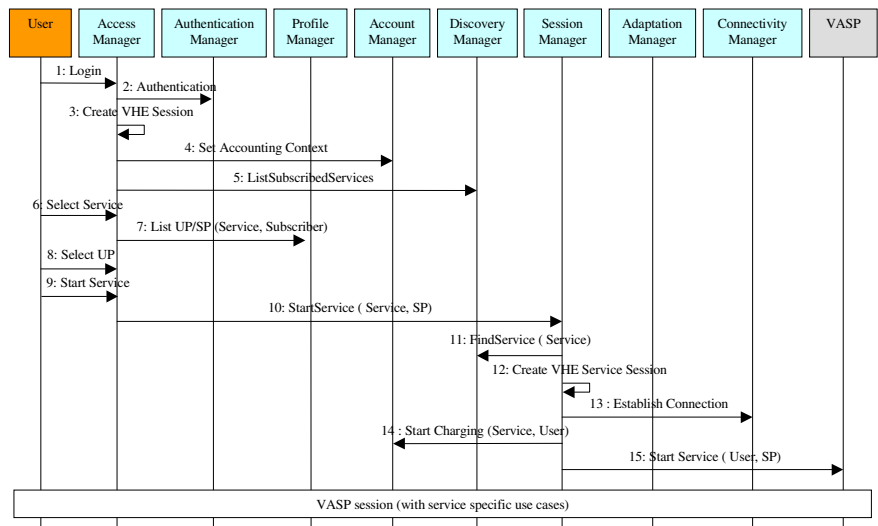


Fig. 3. User login to a VHE Provider and start a VASP service.

Start service is requested to the Session Manager (10). After the Session Manager obtains the service address (11), a VHE Service Session is created (12) and a connection to the VASP is requested (13). The Accounting Manager starts the charging process (14) and a request to the VASP is sent to start the selected service (15). After that a VASP session is created and the specific service use cases take place in the context of this session.

Logout VHE Provider / Terminate Service

The user requests to logout from the VHE Provider (1). The Access component checks if there is any active VHE service session (2). If the list is empty, all the procedures to delete the VHE session and terminate the connection to the VHE Provider are performed by the Access component (9-11).

If there is any active VHE service session is because the user is currently using a VASP service. In this case the user can choose to terminate the service or he can request to suspend the VASP session (described in the next session). If the user chooses to terminate the service (3), the Session component is informed (4) and before the deletion of the VHE Service Session (7), the Session component requests the profiles update (5), in the case some changes had occurred, and requests the Accounting component to stop charging the user for using this service (6).

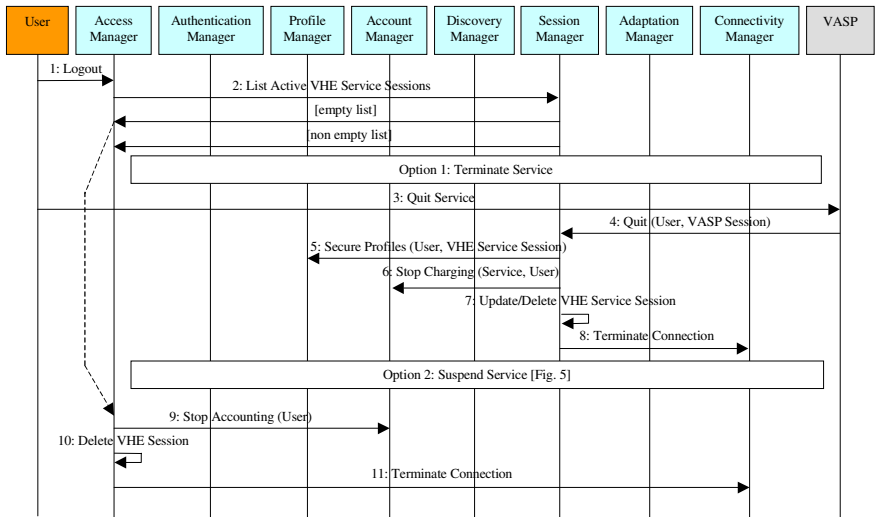


Fig. 4. Logout VHE Provider / Terminate service

After this procedures, the connection between the user and the VASP is terminated (8) as well as the connection between the user and the VHE Provider (9-11).

Suspend / Resume a VASP Session

The user requests the VASP to suspend his participation in the service (1). The VASP session should be secured in such a way that the user can resume his context, including his service preferences, later in time (2). For that, the Session component is informed that the VASP session will be suspended (3) and it requests the Profile component to update the service preferences, if some changes had occurred (4).

The Session component notifies the Accounting component to adapt the charging of the user to his suspended state (5). Accounting may not stop but follow another policy. The Session component updates the VHE service session to the user suspended state (6) and requests to terminate the connection between the user and the VASP.

When the user login again to the VHE Provider and selects to use the suspended service (8) he receives a list of suspended sessions so that he can select one to resume (10). After the user chooses an UP to define an usage context (11), he informs the VHE Provider that he wants to resume a specific VASP session (12).

The Session component then resumes the VHE service session (14), requests the connection to VASP (15), requests the charging resume to Accounting component (16) and informs the VASP to resume the VASP session (17). After that the VASP session is resumed and the specific service use cases take place in the context of this session.

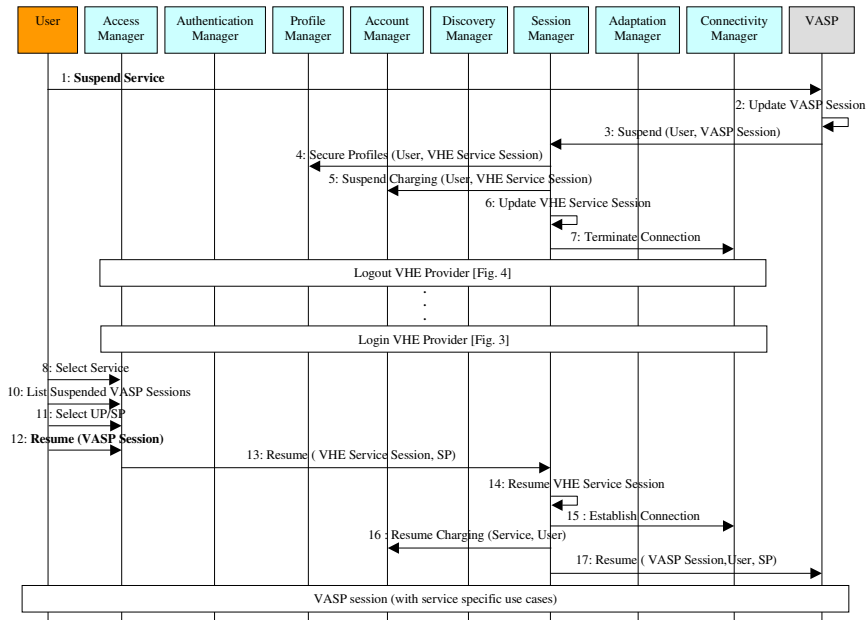


Fig. 5. Suspend / Resume a VASP Session.

5 Two Examples of Applications

INRIA, in France, and Portugal Telecom Inovação with the collaboration of INESC Porto, in Portugal, are partners of the VESPER consortium and are currently working in the development of two services that will be used to validate the project VHE architecture [9]. These services are the Calendar service and the Customer Care service. The develop these two services without a VHE architecture providing basic VHE functionalities would represent a much harder task and the result would be a very difficult to port solution.

5.1 The Calendar Service

The Calendar service provides a coordinating environment allowing multiple users to set up their meetings. In a distributed context, each user has his own calendar that should follow him on travel for his convenience of use. When the Calendar service receives an invitation to a meeting, it must contact each of the invited attendees, specifying the meeting time slot and its location, collect the replies from attendees, decide the meeting time and inform the attendees of the final result.

The server must cope with the fact that different parties involved in the meeting are in different networks, use different access mechanisms and that the dialogue with a number of the parties may take time because either a party is offline or simply cannot

answer immediately. This process can be regarded as the coordination of atomic and consistent updates of distributed heterogeneous databases (agendas).

The service resolves the heterogeneity by adapting dialogues to the attendees' conditions, so that the parties are solicited in their preferred way.

The use of the Calendar service can be envisaged in (almost) any kind of terminals, ranging from a mobile telephone or a PDA to a PC. The service adapts accordingly. For example, a party on a mobile phone can be contacted either with a short text string or with voice, a party connected with a PDA can be contacted by a low quality small image showing the plan to access the meeting point together with text indicating the meeting time slot, whereas a party on PC can be contacted with a full colored "zoomable" image or via e-mail.

A user can specify in his service preferences profile where he wants to be reached depending on the date or the time (for example, at his work PC from 9am to 6pm, at his home fixed phone from 7pm to 10pm) or at certain locations (for example, on his freehand mobile phone while he is driving on the highway).

It may happen that a party cannot answer to the meeting invitation because he is not reachable or cannot reply immediately because, for instance, he is already on the phone or in another meeting. As a result, the final common decision would be delayed. It is assumed that a non-reachable party can, in some occasions, delegate his agenda policy to an agent program so that this agent can reply immediately to the meeting request. The corresponding party is informed of the meeting when he becomes reachable.

A consistent view of the meeting time is guaranteed to all parties despite interruptions of the meeting fixing process and despite failures.

5.2 The Customer Care Service

The Customer Care service intends to give support to customers by means of interactive product tutorials, solutions proposal through question & answers assistance, online-operator assistance and other support services, such as "software patch download".

A real application example for this service could be a software, hardware or electronic equipment company that gives support to the installation, configuration and usage of the company products.

In a usage scenario, after the user login and if successfully authenticated by the service, the Customer Care service offers several functionalities:

- **Product Tutorial:** the user interactively accesses information about products and services from his service/product supplier, organized in a tutorial way. The information is basically organised as a multimedia slide session, where the user can navigate back, forward or repeat a specific slide. The user can also request the help of an online-operator though an audio/video conference.
- **Audio/Video Conference:** the user can request the establishment of an audio/video conference with an online-operator. The Customer Care service should support the establishment of an Audio/Video conference between different types of terminals and over different types of networks. The intention is to have the possibility of establishing an audio conference between two mobile phones, an audio conference between an application in a PC and a mobile phone, a video conference using two PCs or an audio conference between a fixed phone and a mobile phone,

transparently to the user. The audio/video conference connection could be made based upon the location of user and online-operators.

- Questions & Answers: the user has the possibility to follow a questionnaire, answer some key questions, so that the service can conclude about the user's problem and propose a solution.
- Software/Information Download: the user can search and download software patches, product manual pdfs, etc, and the online-operators can obtain information about previous user's sessions.

Besides these basic service specific functionalities, the Customer Care service should also enable the user to personalise his service usage, i.e. change the different profiles "seen" within the service context and suspend a service session that can be resumed later in time, in a different location, with different access network and terminal.

6 Conclusions

Although the Virtual Home Environment has been identified as a powerful and necessary concept for the future integrated service environment, little work has been done so far for its precise definition and validation. The VHE requirements will not be met without defining and specifying an architectural framework, enabling the realisation of a VHE for service provision and use. The VESPER project appeared with this key objective.

The VESPER approach to reach this objective is based on current trends and developments that take place in the context of the 3rd generation networks. In order to take full advantage of network transparency access by service providers, VESPER advocate the use of open APIs like the OSA/Parlay frameworks. In the same way, VESPER defines an open API, available to service providers, that enables personalization, adaptation to terminals or session mobility to services provided in the context of a VHE provider.

This framework provides many benefits to service providers and service developers. The use of these open APIs, with all the advantages of the VHE features, allows easy and rapid development of new sophisticated telecommunication services.

References

1. 3rd Generation Partnership Project (3GPP), <http://www.3gpp.org>
2. 3GPP, 3G TS 22.121 v3.2.0, The Virtual Home Environment, stage 1.
3. VESPER Welcome Page; <http://vesper.intranet.gr/>
4. TINA-C, TINA Consortium, <http://www.tinac.com>
5. Parlay Group, Parlay Specifications, <http://www.parlay.org/>
6. 3GPP, 3G TS 29.198 v3.0.0, Open Service Access (OSA) API Part 1, stage 3.
7. 3GPP, 3G TS 23.127 v3.1.0 Virtual Home Environment / Open Service Architecture.
8. VESPER, IST-1999-10825, D21 - VHE Requirements, Nov. 2000.
9. VESPER, IST-1999-10825, D41 - Demonstration Services Requirements, Jan. 2001.

WLL Link Layer Protocol for QoS Support of Multi-service Internet

Huan Pham¹, Bill Lavery¹, and Hoang Nam Nguyen²

¹ School of Information Technology
James Cook University, QLD 4811, Australia
{huan, bill}@cs.jcu.edu.au

² Institute of Communication Networks
Vienna University of Technology
Favoritenstrasse 9/388, A-1040 Vienna, Austria
namhoang@pop.tuwien.ac.at

Abstract. Wireless Local Loop (WLL) such as LMDS typically has time-varying and high Bit Error Rate (BER) channel characteristics [1-3]. To provide Internet service over a such link, error protection mechanisms at link layer are necessary [4-6]. However, since different services provided by higher layers require different link layer behavior, a multi-service link layer architecture is needed, as has been proposed in [7]. The later assigns transmission bandwidth in proportion to network layer allocation independent of arbitrary overhead rates. As a different approach, we propose a WLL link layer architecture and protocol that assign transmission bandwidths according to services' priorities and their actual bandwidth needs (taking in to account network layer allocation as well as any overhead occurred). This new model keeps track of the current status of each link service, such as link service overhead rates, and informs the network layer. We argue that our link layer architecture and protocol more effectively support QoS services implemented at higher layers, especially those that require hard QoS guarantees (such as guaranteed service in IntServ model [9] and Expedited Forwarding (EF) service in DiffServ [10]).

1. Introduction

A Wireless Local Loop (WLL) channel typically has time-varying and high Bit Error Rate (BER) channel characteristics. Providing Internet service, especially with QoS guarantee, therefore requires additional error protection mechanisms at link layer [4-6]. In addition, the mixed traffic nature of the Internet also requires that different error protection mechanisms be applied to different services. Finally, there should be a cooperative mechanism between link layer and the network layer to effectively control the use of limited link bandwidth resource by different Internet services, each with different link layer overhead. Our proposed link layer architecture and protocol seek to answer these requirements.

The rest of the paper is organized as follows. First we discuss the proposed solutions to solve the issue of providing QoS Internet over time varying, error prone wireless links, highlighting the unsolved problems. We then propose a new link layer architecture and protocol as our solution to solving the QoS problems. We then

consider an example of applying our model for a simple Internet traffic including a real-time UDP traffic and a non-real time TCP traffic. Finally a short summary will conclude the paper.

2. Effect of a Time-Varying BER Link on Multi-service Internet

Currently much of the Internet services are built on the traditional TCP/IP protocol stack. Transport layer protocol TCP was designed with the observation that most losses on the Internet are due to congestion, as routers run out of buffers and discard incoming traffic. As a consequence, when running TCP over lossy links, TCP assumes all the losses are due to network congestion and invokes its congestion avoidance mechanism, resulting in a degraded throughput performance.

Schemes to improve TCP performance over error prone links have been well studied and summarised in [4-6]. According to [4], these schemes can be divided into three categories: end-to-end protocols (modifications made to TCP itself); link-layer protocols that provide reliability at the link layer; and split connection protocols that break the end-to-end connection into two parts at the base station. All these studies [4-6] show that link layer schemes are easier to implement and provide a better performance than the other two options.

However it is expected that Internet traffic is and will always be of mixed origins: non-real-time traffic usually implements TCP as transport layer; while real time traffic may use UDP. The real-time traffic can usually accept some level of packet loss while it is more critical with respect to timely packet deliveries. Forward Error Correction (FEC) is probably a good option for much of the real time traffic. The non-real time traffic, on the other hand, is more critical with respect to packet loss rate while it can tolerate variable delay. For the latter, error protection link layer mechanisms such as retransmissions with a local timeout are frequently used. If the wireless link layer is to be able to simultaneously handle both types of traffic, it should be able to differentiate the traffic introduced from the higher layers and provide each with a suitable QoS behavior. In other words, a multi-service link layer is needed.

Reference [7] introduced a multi-service wireless link layer architecture. The model tries to isolate the different services and to prevent one from interfering with the others. According to the model in [7], the link layer should allocate link layer bandwidth in the proportions presented to it by the network layer traffic requirements, independent of link layer overhead rates. However, there are two issues associated with this approach.

The first results from the interaction between the rate adaption mechanism employed by the TCP transport layer protocol and the link layer allocation strategy of [7]. When there is unused bandwidth, the TCP layer will ramp up its throughput, to better utilise the available bandwidth. However the increasing transport layer throughput will be reflected in an increasing Network layer traffic demand to the link layer. In this scenario, with the allocation strategy of [7], the TCP stream will then be allocated a progressively increasing share of the link layer bandwidth. With TCP overhead rates lower than UDP overhead rates and with link layer allocation proportional to network layer traffic demand, the TCP stream will continue to

increase its share of link layer bandwidth, at the expense of the UDP stream. The TCP stream will build till it ‘hogs’ the link layer bandwidth from the required UDP share, and eventually the TCP stream will annexe almost all of the link layer bandwidth. The non rate adaptive higher overhead UDP traffic will then have completely lost its link layer bandwidth allocation.

The second issue is that, the link model does not implement any prioritization at the link layer and relies on higher layers to do this. As a consequence, when the wireless link experiences severe interference, and when the effective link layer throughput drops, the performance of all services are equally degraded, before the link status is updated and the higher layers reschedule services accordingly.

To solve these two issues, while considering the three requirement mentioned in the Introduction of the paper, we propose a new link layer architecture and protocol that take into account differences in service overheads, and allow different services to have different priorities.

3. Proposed Link Layer Protocol

The architecture for the new link protocol is shown in Figure 1. Packets from the network layer introduced to the link layer are first classified and mapped into different link layer services based on their protocol fields (TCP or UDP ports), or based on Type of Service/DiffServ Code Point field (if DiffServ is provided by the network layer). The link layer then does packet fragmenting. Error protection service such as Forward Error Correction or Retransmission are implemented, and can be implemented differently for different services. Link layer data units (frames) are scheduled for transmission using a self-clocked Weighted Fair Queuing Scheduler (WFQS) with different priorities of being dropped in case of link difficulty. At the receiver side, frames are demultiplexed, reassembled and introduced back to the network layer.

Compared to [7], two new function blocks are introduced in our model: allocation measurement by the network layer and link layer bandwidth consumption measurement. The allocation measurement block measures the traffic assigned by the network layer for different service classes. The consumption measurement block measures the actual bandwidth required at the link layer for each service type. That bandwidth includes the network layer payload as well as the overhead (FEC or Retransmission) added by the link layer.

These two new function blocks aim to keep track of the percentage of the overhead added by each link layer service. These overhead rates vary according to link state if retransmission or variable rate FEC are used for error protection; the higher BER is, the higher the overhead rate must be. The link layer will provide feedback to the network layer, informing it of the maximum network layer throughput or “normalized capacity”, C_i , for each service if that service was operative alone. This “normalized capacity” is the total link capacity, C , divided by a factor of $(1 + \alpha_i)$, where α_i is the link layer overhead rate for service i .

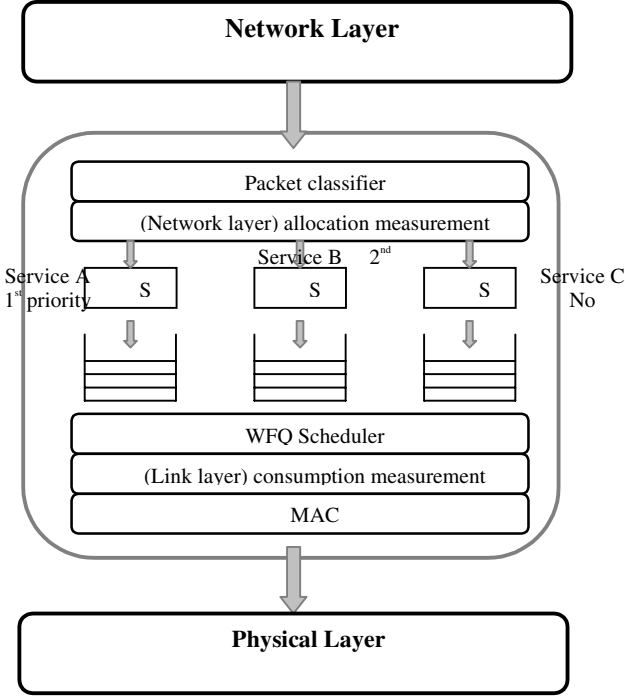


Fig. 1. Multi-service QoS link layer architecture

A Self-Clocked Weighted Fair Queuing Scheduler (WFQS) schedules the next frames according to the service weights; these weights are equal to the desired raw link capacity for each service, i.e. the product of allocation bandwidth measured during the last time interval, and $(1 + \alpha_i)$. In general, a higher value of α_i implies a higher priority. When the link capacity decreases (for example due to the change of modulation scheme), indicating the chance of frame dropping, the frames with lowest priority will be dropped first before any higher priority frames. In other words, this model allows services with a higher priority to take the bandwidth from lower priority services. Therefore, the higher priority services stand a better chance of not being affected by a temporary bad channel condition. Please note that this frame dropping is a temporary measure by the link layer, in order to protect frames of higher priority from being dropped. For longer term, the problem should be solved by network layer packet scheduling adjustment, when it gets the link layer state update at the end of the next time interval. This protocol works as follows:

If the total link bandwidth is C , normalised link layer capacity for the service i is C_i , where

$$C_i = \frac{C}{1 + \alpha_i} \quad (1)$$

and the allocation bandwidth introduced by the Network Layer for that corresponding service is W_i , then in order not to over-assign the link capacity, the allocation bandwidths and the normalised link capacities should meet the following requirement:

$$\sum_i \frac{W_i}{C_i} \leq 1 \quad (2)$$

If the requirement (2) is not satisfied, the network layer knows that the link layer is having difficulty supporting the current bandwidth allocation. The network layer will adjust the current bandwidth allocation, for example, by dropping TCP packets, and thereby triggering the congestion avoidance mechanism at transport layer, or by notifying the application of the real-time traffic to use a lower rate option.

In general, the network layer has better knowledge of the original traffic characteristics and can make a better decision in case of link difficulty, rather than just leaving the link layer to drop the performance of all services equally. The service with highest priority at the link layer can be mapped to guaranteed traffic (Expedited Forward Behavior) provided by DiffServ [9], guaranteed flows in IntServ model [10], or realtime UDP traffic in the current best effort Internet model. Service with lowest priority can be mapped to Best Effort traffic in the DiffServ model or in the IntServ model, or traffic such as FTP, E-mail, in current Internet. Different priorities can also be used to differentiate different users as well.

In a dynamic bandwidth assignment environment, different users share the total bandwidth dynamically based on each user's current requirement. For such an environment, an example of a suitable QoS medium access control (MAC) for WLL is introduced in [8]. Our model, with different service queues and different priority levels, creates the platform for supporting such an implementation. Then the bandwidth is fairly shared, without the danger of the bandwidth hogging detriment of the model of [7].

4. A Mixed Service Example

In this part, let's consider an example of mixed real-time and non-real-time traffic; real-time traffic uses TCP as its transport protocol and non-real-time traffic uses UDP. The following assumptions are made:

- Channel capacity: 1 Mb/s
- Non-real time offered traffic: 300Kb/s,
- Real-time offered traffic: 200Kb/s
- Non-real-time traffic uses retransmissions as a error protection mechanism, with a maximum number of retransmissions of 5. Average packet size for non-real-time traffic is 1000 bit/packet.
- Real-time traffic uses FEC, with 100% overhead.
- BPSK modulation scheme with coherent detection

First we estimate the overhead rate for the TCP traffic due to error induced retransmission. According to [11], the relationship between the BER of the channel, P_b , and E_b/N_0 is:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (3)$$

Therefore non-real-time frame error rate P_f is

$$P_f = 1 - (1 - P_b)^{1000} \quad (4)$$

The probability of a frame being retransmitted for k time ($0 \leq k \leq 4$) will be:

$$P_r(k) = (1 - P_f) \times (P_f)^k \quad (5)$$

and probability of a frame being retransmitted 5 times (maximum) will be:

$$P_r(5) = (P_f)^5 \quad (6)$$

From equations (3-6), a relation between the percentage of overhead for the non-real-time traffic using retransmission can be established using the following equation:

$$\alpha = \sum_{k=0}^5 P_r(k) \times k \quad (7)$$

The overhead for the real-time traffic is 100% and is unchanged despite the channel characteristic variability, because it uses FEC for error protection. In practice, these service overhead rates are achieved by using the network layer traffic allocation measurement and link layer bandwidth consumption measurement, and are provided to the network layer regularly.

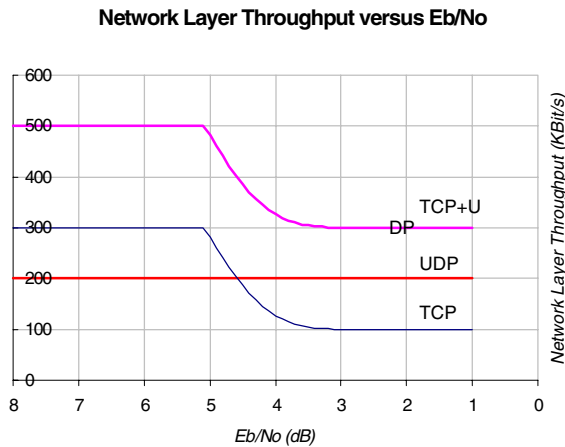
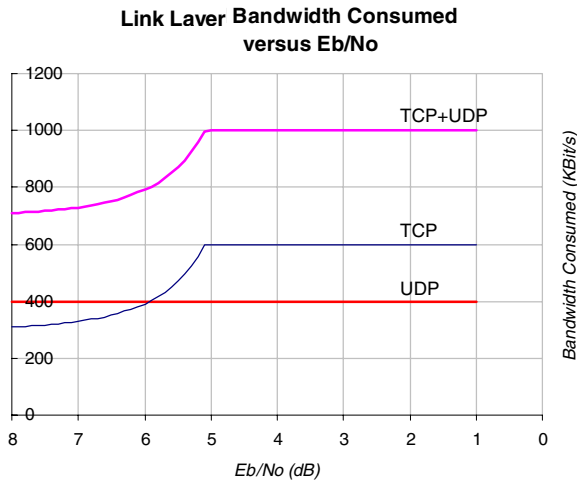
The two graphs below illustrate how network layer throughput and link layer bandwidth consumed by each of the TCP and UDP based services varies with channel characteristics (BER). The graphs show that, when the channel has an E_b/N_0 greater or equal than 6dB, the 300 Kb/s of TCP traffic consumes just over 300 Kb/s of link layer bandwidth, while the 200 Kb/s of UDP traffic with its 100% FEC overhead consumes 400 Kb/s of link layer bandwidth. Eventually, when the E_b/N_0 drops below 5 dB, note that it is the TCP traffic which experience a drop in network layer throughput. Our protocol degrades throughput of the lower priority TCP traffic but maintains throughput of the higher priority UDP traffic.

Note however that, unlike [7], because our protocol takes in to account link layer overhead in allocating link layer bandwidth, our protocol does not suffer from bandwidth hogging due to higher layer rate adaptation mechanisms.

5. Conclusion

In this paper we have proposed a new QoS supported multi-service link layer architecture and protocol, which is superior to that proposed in [7]. Our link layer model tries to preserve bandwidth allocation from the higher layer, while considering the overhead introduced by the link layer itself. Even in the case of severe interference, the link layer can still support hard QoS requirement for the more important services while degrading the performance of lower priority services. This is critical to support new QoS Internet services such as IntServ or DiffServ over highly unpredictable error prone links. We have also provided an example of carrying mixed

real-time and non-real-time traffic over the proposed link layer architecture and analysed its behaviour. In ongoing research, we are implementing the proposed multi-service link model, for a range of traffic streams typical of a wide area ISP network.



Acknowledgement

We would like to thank A/Prof. Greg Allen for his very useful comments and G. Xylomenos and G. C. Polyzos - the authors of [7] - for the discussions and clarifications on their paper. This work has been supported by James Cook University through an International Post-Graduate Research Scholarship.

References

1. V.I. Roman: Frequency Reuse and System Deployment in Local Multipoint Distribution Service. IEEE Personal Communications, December 1999.
2. G.M Stamatelos and D. D. Flaconer: Millimeter radio access to multimedia services via LMDS. Global Telecommunications Conference proceedings, 1996, pages: 1603 -1607 vol.3
3. N. E. Daly et al: Capacity effects on terrestrial broadband wireless access networks, operating in the LMDS frequency band, during rainfall conditions. IEEE Vehicular Technology Conference proceedings, 2000, pages: 2444 -2448 vol.3
4. H. Balakrishnan et al: A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. IEEE/ACM Transactions on Networking, December, 1997, Vol. 5, No. 6.
5. H. M. Chaskar et al: TCP over Wireless with Link Level Error Control: Analysis and Design Methodology. IEEE/ACM Transactions on Networking, October 1999, Vol. 7, No. 5.
6. G. Xylomenos and G. C. Polyzos: Internet Protocol Performance over Networks with Wireless Links. IEEE Network, July/August 1999.
7. G. Xylomenos and G. C. Polyzos: Link layer support for Quality of Service on Wireless Internet Links. IEEE Personal Communications, October 1999.
8. A. Baiocchi et al: A MAC Protocol supporting IP with QoS Guarantees in a wireless local loop. Proceeding of Global Telecommunications Conference, 1999.
9. S. Blake: An Architecture for Differentiated Services. RFC 2475, December 1998
- 10.R. Braden: Integrated Services in the Internet Architecture: an Overview. RFC 1633, June 1994.
- 11.B. Sklar: Digital Communications. Prentice Hall, US, 1988.

Biographies

Huan Pham holds a bachelor degree in engineering (telecommunications) from Hanoi University of Technology, Vietnam, a master degree in engineering (telecommunications) from Swinburne University of Technology, Australia. He is currently a PhD student at School of Information Technology, James Cook University, Australia. His research interests include QoS over wireless links, and network dimensioning to support QoS.

Prof. Dr. Bill Lavery is the head of school of Information technology. Prof. Lavery's current research interests are in the field of convergent technologies - applications and access technologies in the converging Internet, telecommunications, computer networking, media and multimedia fields, including mobile & wireless communications technologies.

Hoang Nam Nguyen holds a bachelor degree and master degree in engineering (telecommunications) from Hanoi University of Technology, Vietnam. He is currently a PhD student at the Institute of Communication Networks, Vienna University of Technology, Austria. His research interests include QoS for wireless and satellite communications.

The Earliest Deadline First Scheduling with Active Buffer Management for Real-Time Traffic in the Internet

Xiaojun Hei and Danny H.K. Tsang

Dept. of Electrical & Electronic Engineering
Hongkong University of Science and Technology
Clear Water Bay, Kowloon, Hongkong
`{heixj, eetsang}@ee.ust.hk`

Abstract. We study the problem of QoS guarantee for differentiated services. A two-level hierarchical scheduling framework is deployed for the separation of the QoS metrics. Due to the desirable property of minimizing the maximum packet lateness, the Earliest Deadline First (EDF) scheduling is adopted to provide the in-class scheduling for the time-sensitive traffic. We propose to employ an EDF scheduler combined with an active buffer management scheme (CHOKe) to improve the fairness of resource allocation and to maintain a good delay performance for all real-time applications. Simulation results show that the proposed scheme can achieve a better delay performance and make a more fair bandwidth allocation between the real-time TCP and UDP connections than the First Come First Serve (FCFS) scheduling with the Drop-Tail buffer management which is commonly deployed in the traditional IP router.

Keywords: Scheduling, Earliest Deadline First, Active Buffer Management, Real-time Traffic

1 Introduction

With the development of digital technology, the network convergence is occurring at both the media and the technological levels. The recent achievements in the fiber technology of Wavelength-Division Multiplexing (WDM), promise a large amount of bandwidth in the future high-speed networks, so that it is possible for the multimedia applications to run together with the traditional data-oriented services within a single common network. The Internet turns out to become the dominant networking technology, yet the multimedia communication with Quality-of-Service (QoS) guarantee drives the need of the substantial changes in the current Internet infrastructure.

The multimedia services, such as voice, video and other applications, demand not only high bandwidth, but also a stringent real-time delay constraint due to the fact that the value of the communication depends upon the time at which messages are successfully delivered to the recipient. The above services are

commonly classified as either soft or hard real-time applications. Soft-real-time applications can tolerate some amount of lost messages, while hard-real-time applications have zero loss tolerance [3].

Internet telephony is one of the promising soft-real-time applications, which can permit a few lost phonemes (critical units of speech) in a continuous speech [10]. On the other hand, in the web-based visualization application, large data sets are processed in the central server and the results need to be transferred back to the client sides for visualization [11]. Due to the interactive nature of the whole process, both stringent delay and loss-free data transmission are required. Further, the dominant application running on the current Internet is World Wide Web (WWW). WWW can support a rich palette of media types, such as pictures, audio and video, instead of the past text-only Internet. Web surfing is a highly interactive activity; there are not only lossless data communication but also lossable multimedia communication involved.

Because of the tight latency bound in the above multimedia applications, it is suggested that they had better not use a reliable transport like TCP because retransmission of the lost packet may probably cause packets to arrive too late to be useful [1]. Instead, the unreliable transport, such as UDP, is recommended. However, the hard-real-time application requires both stringent delay and zero packet loss, so if it is built up based on an unreliable transport, the application must deal with the missing packets, therefore the complexity increases in the application software. The recent adopted Real-Time Transport Protocol (RTP) follows this approach, which provides mechanisms for dealing with impairment such as jitter and loss, as well as for timing recovery and intermedia synchronization. In the recent proposed framework of Differentiated Service (DiffServ), besides the traditional best-effort traffic, two other traffic classes with different Per-Hop-Behavior (PHB), Expedited Forwarding (EF) and Assured Forwarding (AF), have been suggested to serve real-time applications. The transport in the framework is not clearly defined yet, so it is interesting to see whether the hard-real-time application can run over TCP. In this paper, the soft-real-time and hard-real-time applications are proposed to run on UDP datagrams and TCP flows respectively, and based on the proper scheduling and buffer management schemes, they can be better served in terms of delay and packet loss.

In order to support communication service with the QoS guarantee, the network resources need to be managed in a systematic manner. The FCFS-DropTail scheme, which is commonly implemented in traditional IP routers, has some problems in the QoS provisions. The FCFS policy can achieve a tight delay bound by limiting the buffer size, then packets will be dropped with a larger probability when they arrive at routers and there are not enough buffer to store them. The situation may be even worse if the traffic is highly bursty. On the other hand, the Drop-Tail buffer management can result in global synchronization among multiple TCP connections, which can underutilize the congested link because several connections may halve their congestion window at the same time. Moreover, the FCFS-DropTail scheme cannot implement the fair resource allocation among TCP and UDP flows [12].

In the past several years, new scheduling schemes are proposed. They are basically the variants of two fundamental disciplines, Generalized Processor Sharing (GPS) and Earliest Deadline First (EDF). The packetized version of the GPS scheduler (PGPS) can guarantee the minimum per-connection throughput and delay bound with flow protection, but it needs to maintain per-connection states and reserve large bandwidth for a small delay bound. With the desirable property for the EDF scheduler to minimize the maximum lateness of packets [9], the EDF takes advantage over the PGPS to schedule the real-time traffic in terms of system scalability and utilization. On the other hand, TCP performance can be improved by the active queue management [7], such as Random Early Detection (RED), in which the average queueing delay can be controlled while the transient queue-size fluctuation is allowed. However, like Drop Tail, RED is also unable to penalize unresponsive flows [12]. Recently, embedded in RED a stateless active queue management scheme, with the name of CHOKE, is proposed to work with the FCFS scheduling for approximating fair bandwidth allocation, which tries to bridge fairness and simplicity in the scheduler [12].

A lot of studies have been done upon the scheduling schemes and the buffer management schemes respectively. In fact, the queue management strategy can be used along with any scheduling scheme, thus it is suggested to study them in an integrated fashion in order to get the optimal performance [8]. In this paper, the EDF scheme is proposed to schedule the real time traffic in order to get the optimal delay performance whereas the active buffer management CHOKE works together with the above EDF scheduling to improve the fairness of resource allocation between UDP-based and TCP-based real-time communications.

The paper is structured as follows. In section 2, we introduce the 2-level hierarchical scheduling framework in order to support differentiated traffic in the Internet, and describe how the EDF scheduler cooperates with the active buffer management scheme (CHOKE). Then the simulation based performance evaluation is carried out in section 3. Finally we conclude the paper and present the future work in section 4.

2 Hierarchical Scheduling Framework

The hierarchical scheduling aims to meet the goals of sharing link capacity and providing differentiated service, such as real-time service, best-effort service, and others [6]. The link-sharing is first proposed in [4], in which network resources are shared among traffic streams and they are grouped according to administrative affiliation, protocol, traffic type, or other criteria. The concept of link-sharing is implemented as a particular resource management scheduling scheme called Class Based Queueing (CBQ). In CBQ, the user traffic is organized into a tree, or hierarchy, of classes, and traffic classes are differentiated by the network. The FCFS-DropTail scheme is still suggested to serve the in-class packets due to its simplicity. Obviously, the bandwidth allocation is the major concern in CBQ, however, multimedia applications also require a tight delay bound. Thereafter, the fluid Hierarchical Generalized Processor Sharing (H-GPS) system is proposed

as a general and flexible framework to support hierarchical link sharing and traffic management for different classes [6]. The H-GPS scheme provides a more fine-grained link-sharing structure, which can provide a guaranteed end-to-end delay bound for a session if the traffic in that session is leaky-bucket constrained, however, this delay bound is very conservative. On the other hand, without a large bandwidth reservation in EDF, applications are still possible to obtain a small delay bound by assigning packets with more urgent time-tags as their deadlines. So in this paper the EDF scheduler or its variants are proposed to replace the corresponding GPS scheduler for time-sensitive traffic in the H-GPS framework.

Based on the above discussions, the following hierarchical scheduling framework is introduced in order to fairly allocate bandwidth among classes in the higher level by the GPS scheduler and to maintain the particular throughput or delay QoS guarantee by the proper selection of the GPS or EDF scheduler in the lower level (see Figure 1). Due to the scalability concern, the scheduling framework tries to provide service with a class-based QoS. There are two types of schedulers in the framework, the general scheduler and the link-sharing scheduler. The link-sharing scheduler allocates bandwidth among classes and the general scheduler tries to serve a traffic class with its allocated bandwidth share. They determine exclusively the packet scheduling in the absence of congestion. In the presence of congestion, the link-sharing scheduler controls the scheduling of the packets from different classes. The GPS link-sharing scheduler can ensure that each interior or leaf class in the multilevel structure will receive its allocated bandwidth over appropriate time intervals, and distribute any “excess” bandwidth fairly among classes.

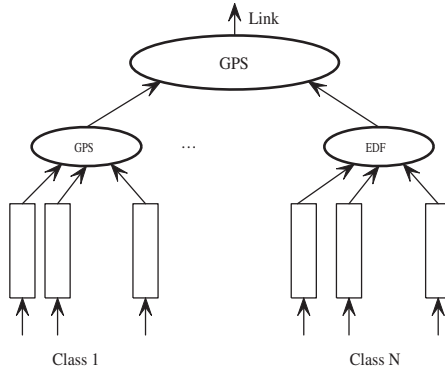


Fig. 1. Hierarchical Scheduling Framework

The traditional best-effort traffic will still run on the future Internet and there may be more services emerging. In this paper, we are only considering how to provide service for the best-effort and real-time traffic in the two-level

hierarchical scheduling framework, which can be depicted in Figure 2. If one more traffic class appears, it should be easy to add one more branch in the hierarchical scheduler tree.

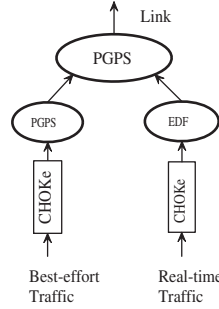


Fig. 2. Hierarchical Scheduling Structure

To simplify the problem of how the EDF scheduler cooperates with the CHOke scheme, we focus our attention only on the EDF-CHOke branch in the tree. The real-time traffic class can be guaranteed with the worst case of minimum bandwidth by the upper PGPS scheduler. If the best-effort traffic has not used up its bandwidth, the PGPS scheduler can allocate this excess bandwidth to the real-time traffic, therefore the QoS provision for the real time traffic will be better.

In the later section, the comparison studies are conducted for the various combinations between the service policies (FCFS and EDF) and buffer management schemes (DropTail, RED and CHOke) in the simulation experiments.

3 Performance Evaluation

3.1 Simulation Configuration

The simulations are carried out in the Network Simulator [13]. Consider the following simulation scenario with a single congested link, as shown in Figure 3 to study how much bandwidth a single nonadaptive UDP source can obtain when routers use different schemes. The congested link in the network is between the routers R1 and R2. The link, with the capacity of 1 *Mbps*, is shared by 1 UDP and 32 TCP flows. Each source and destination node is connected to the router using a 10 *Mbps* link, which is ten times the bottleneck link bandwidth. All the links have a small propagation delay of 1 *ms* so that the delay experienced by the packet is mainly caused by the queueing delay in the buffer rather than the transmission delay or propagation delay. Different scheduling and buffer management schemes are deployed in the congested link for the comparison studies. The maximum window size of TCP is set to 300 such that it does not

become the limiting factor of the TCP flows' throughput. The TCP flows are derived from FTP sessions which transmit a very large size file and the UDP source sends packets at a constant bit rate of 2 *Mbps*, so the link between the routers R1 and R2 becomes the bottleneck link in the network. All the packets are assumed to have the same fixed size of 1000 bytes.

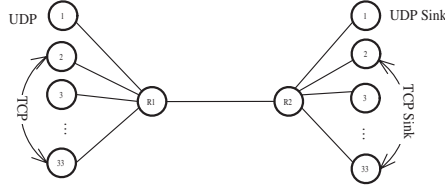


Fig. 3. Network Topology

The FCFS-DropTail, FCFS-RED, FCFS-CHOKe, EDF-DropTail, EDF-RED and EDF-CHOKe schemes are studied for comparison. The minimum threshold min_{th} in both RED and CHOKe is set to 100 packets, and the maximum threshold max_{th} is set to be twice the min_{th} . The physical queue size in the above schedulers is set to 300 packets. The delay requirements for both UDP and TCP flows are set to 600 *ms*.

3.2 Simulation Results

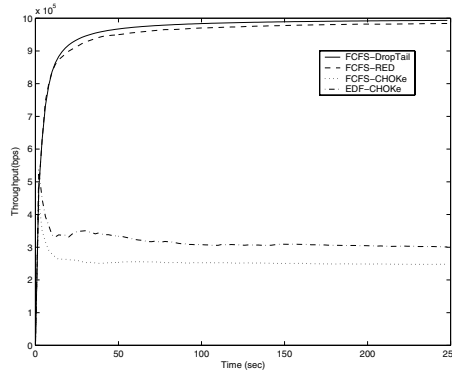


Fig. 4. UDP Throughput Comparison

The throughputs of the UDP flow under different schemes: FCFS-DropTail, FCFS-RED, FCFS-CHOKe and EDF-CHOKe are plotted in Figure 4. From Figure 4, it is clearly shown that the FCFS-DropTail and FCFS-RED schemes do

not discriminate against the unresponsive UDP flow. The UDP flow takes away more than 95% of the bottleneck link capacity and all the TCP connections can only share the remaining 5% bandwidth. The FCFS-CHOKe scheme provides a fairly good resource allocation, in which the total TCP goodput takes up around 750 *Kbps*, while the EDF-CHOKe scheme has a little less TCP goodput than CHOKe, which is about 700 *Kbps*. The individual throughputs of the 33 connections in the above FCFS-CHOKe and EDF-CHOKe schemes along with their ideal fair shares are plotted in Figure 5.

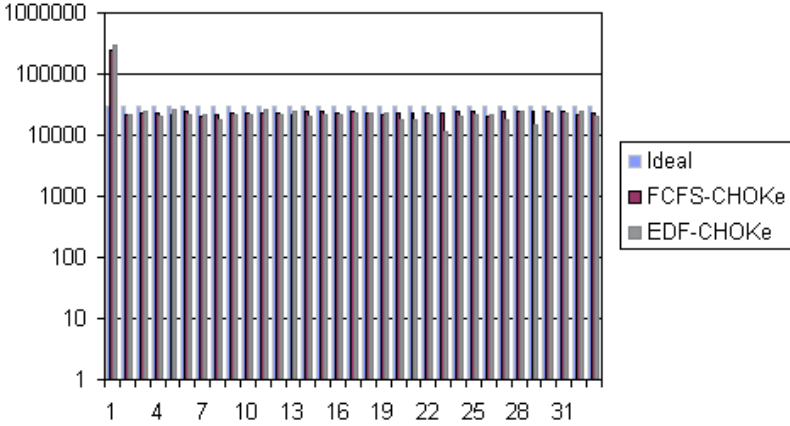


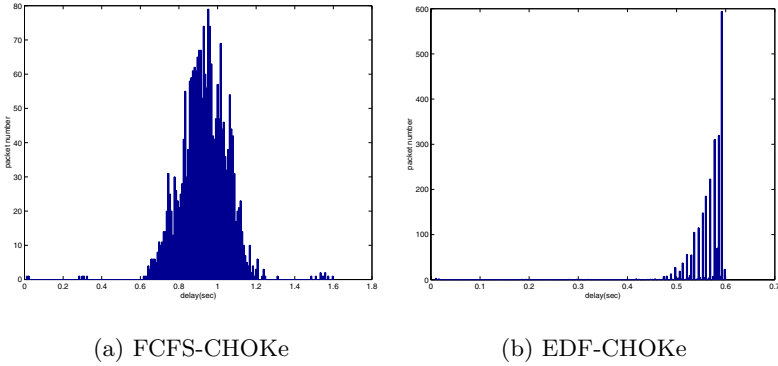
Fig. 5. Per Flow Throuput Comparison

To provide a quantitative comparison, we adopt the concept of the fairness index [2]. The fairness index always results in a number between 0 and 1, with 1 representing the greatest fairness. Based on the results in Table 1, we can see that both FCFS and EDF working together with DropTail and RED cannot provide a fair bandwidth allocation. Though the EDF-CHOKe scheme is not more fair than the FCFS-CHOKe scheme, it is shown that with the active buffer management like CHOKe, the EDF scheduling can have much better fairness than the traditional EDF with the Drop-Tail buffer scheme and the EDF with the active buffer management of RED.

the packet delay distribution at the the congested link between R1 and R2 of the TCP connections in the FCFS-CHOKe and EDF-CHOKe schemes are plotted in Figure 6. Because the FCFS-CHOKe scheme makes the scheduling decision without considering the delay constraint for the communication, most of the TCP packets in EDF-CHOKe are transmitted within their delay constraint (0.6 *sec*) while most of the TCP packets in FCFS-CHOKe suffer deadline violation.

Table 1. Fairness Index Comparison

	Fairness Index
Ideal	1.0
FCFS-DropTail	0.0305
FCFS-RED	0.0309
FCFS-CHOKe	0.3744
EDF-DropTail	0.0304
EDF-RED	0.0304
EDF-CHOKe	0.2838

**Fig. 6.** Packet Delay Distribution of the TCP Connections

Further more, the statistics about the packet delay in the congested link between R1 and R2 for the FCFS-CHOKe and EDF-CHOKe schemes are tabulated in Table 2. As seen clearly in the table, due to the aggressive nature to capture the bandwidth over TCP flows, UDP traffic has a pretty good delay performance, while TCP traffic obtains a very different treatment in terms of delay in the above two schemes. In Table 2, *avg* refers to the mean of packet delay and *std* refers to its standard deviation.

We have investigated the effectiveness of the above scheduling schemes in terms of throughput and delay separately, however, these two principal network metrics are closely related. To describe this relationship, we adopt the *power* of the network [2], which is the ratio between the throughput and the delay. The powers for the TCP and UDP connections in the FCFS-CHOKe and EDF-CHOKe schemes are shown in Table 3. Note that the powers of the TCP connections are calculated with the method of the mean of ratio introduced in [2]. Generally, it is expected to get as much throughput and as little delay as possible, so with a higher power index, the scheme is more effective. Table 3 shows that the powers for both TCP connections and UDP connection in EDF-CHOKe are larger than those in FCFS-CHOKe.

Table 2. The Delay Statistics (sec)

	Total TCP Traffic		UDP Traffic	
	<i>avg</i>	<i>std</i>	<i>avg</i>	<i>std</i>
FCFS-CHOKe	0.9378	0.1203	0.0859	0.2743
EDF-CHOKe	0.5665	0.0398	0.0476	0.1564

Table 3. Power Comparison

	TCP connections	UDP connection
FCFS-CHOKe	2.45×10^4	3.44×10^6
EDF-CHOKe	3.78×10^4	7.65×10^6

Based on Figure 4, Figure 6, Table 1, and Table 2, we can see that the EDF-CHOKe scheme can maintain a good delay performance as well as make a more fair bandwidth allocation between real-time TCP and UDP connections.

Although the EDF-CHOKe scheme has a better performance than the traditional FCFS-DropTail scheme, the former is more complex than the latter in terms of implementation and control. The complexity arises because the EDF-CHOKe scheme has to select the packet with the smallest deadline for transmission on the link. The scheduler needs to maintain a priority list of deadlines and the insertion or deletion from this list has a complexity of $O(\log K)$ operations, where K is the number of packets awaiting transmission. Besides, the CHOKe buffer management will also add more operation complexity due to the necessity of calculating the average queue length for dropping decisions whenever a new packet arrives.

4 Conclusion

With more and more real-time multimedia applications running on the current Internet, the next generation Internet is expected to support a wide range of applications with heterogeneous QoS requirements. In this paper, the hierarchical scheduling framework is proposed in order to maintain a particular throughput or delay QoS guarantees for multimedia applications by building up the two-level hierarchical scheduling structure with the inter-operation between the GPS scheduler in higher level and the EDF scheduler in lower level.

In the above framework, the Earliest Deadline First (EDF) scheduler is proposed to schedule the real-time traffic. Simulation results show that the proposed EDF scheduler working with the active buffer management scheme can achieve a better delay performance and at the same time make a more fair bandwidth allocation between real-time TCP and UDP connections than the First Come First Serve (FCFS) scheduler with the Drop-Tail buffer management.

There are still many interesting issues for future study. First, how to develop the theoretical model for the EDF scheduling so that it can cooperate with the GPS scheduler in the higher hierarchy, which may result in a better traffic management scheme. For example, if there are too many overdue packets in a real-time traffic class, it may be necessary to increase this session weight in the GPS scheduler so as to increase its received bandwidth. Second, in the simulation study, the UDP traffic model is assumed to be the constant bit stream, but recent studies show that the Internet traffic is self-similar [5]. Besides, the proposed scheduling schemes are only studied in a simple network scenario of one congested link. We also intend to explore these mechanisms in more complicated scenarios with multiple congested links.

References

1. Peterson, Larry L., Davie, Bruce S.: Computer networks : a systems approach. 2nd edition, San Francisco : Morgan Kaufmann Publishers, 2000.
2. Jain, Raj: The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modeling. New York : Wiley, 1991.
3. Aras, C.M., Kurose, J.F., Reeves, D.S., Schulzrinne, H.: real-time communication in packet-switched network. Proceedings of the IEEE, vol.82, no.1, pp.122-39, Jan. 1994.
4. Floyd, S., Jacobson, V.: Link-sharing and resource management models for packet networks, IEEE/ACM Transactions on Networking, vol.3, no.4, pp.365-86, Aug. 1995.
5. Paxson, V., Floyd, S.: Wide area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking, vol.3, no.3, pp.226-44, June 1995.
6. Bennett, Jon C. R., Zhang, Hui: Hierarchical packet fair queueing algorithms, IEEE/ACM Transactions on Networking, vol.5, no.5, pp.675-89, Oct. 1997.
7. Braden, B., et al.: Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC 2309, Internet Engineering Task Force, April 1998.
8. Suter, B., Lakshman, T.V., Stiliadis, D., Choudhury, A.K.: Buffer management schemes for supporting TCP in gigabit routers with per-flow queueing. IEEE Journal on Selected Areas in Communications, vol.17, no.6, pp.1159-69, June 1999.
9. Guerin, R., Peris, V.: Quality-of-service in packet networks: basic mechanisms and directions. Computer Networks, 31 (1999) p.169-189.
10. Hassan, M., Nayandoro, A., Atiquzzaman, M.: Internet telephony: services, technical challenges, and products. IEEE Communications Magazine, vol.38, no.4, pp.96-103, April 2000.
11. Bender, M., Klein, R., Disch, A., Ebert, A.: A functional framework for Web-based information visualization systems. IEEE Transactions on Visualization & Computer Graphics, vol.6, no.1, pp.8-23, Jan.-March 2000.
12. Pan, R., Prabhakar, B., Psounis, K.: CHOKe: a stateless active queue management scheme for approximating fair bandwidth allocation. Proceeding of INFOCOM'2000, March 2000.
13. NS (Network Simulator): <http://www.isi.edu/nsnam/ns/>.

Pricing and Provisioning for Guaranteed Internet Services

Zhong Fan

School of Electrical and Electronic Engineering
University of Birmingham
Edgbaston, Birmingham B15 2TT, UK
`z.fan.1@bham.ac.uk`

Abstract. As the Internet evolves into a global commercial infrastructure, there is a growing need to support more enhanced services than the traditional best-effort service. This paper describes an architecture of provisioning guaranteed Internet services. Unlike conventional QoS mechanisms such as Intserv and Diffserv, this architecture enables Internet resource management through market forces, e.g., pricing. A detailed analysis of two scenarios of the underlying IP network has been presented: ECN capable and Diffserv. We concentrate on the pricing model, its implications for traffic management (e.g., admission control) and possible example services. A simulation framework is also discussed.

1 Introduction

The current Internet is based on the so-called best-effort model where all packets are treated equally and the network tries its best to achieve reliable data delivery. Although this simple model is very easy to implement, it has a number of undesirable consequences when the Internet is evolving towards a multi-service network with heterogeneous traffic and diverse quality of service (QoS) requirements. The flat rate charging structure attached to the best-effort model has undoubtedly contributed to the growing problem of congestion on the Internet. Since the network resources are completely shared by all users, the Internet tends to suffer from the well-known economic problem of “tragedy of the commons”. The greedy users will try to grab as much resources as possible, leading to an unstable system and eventually congestion collapses. The Internet has been successful till now because most end systems use TCP congestion control mechanisms and back off during congestion. However, as the number of TCP-unfriendly users increases, such dependence on the end systems’ cooperation is becoming unrealistic. The lack of explicit bandwidth policing and delay guarantees in the current Internet also prevents Internet service providers (ISP) from creating flexible packages to meet the different needs of their customers [1].

As the Internet evolves into a global commercial infrastructure, there is a growing need to support more enhanced services than the traditional best-effort service. To address this issue, there have been intensive efforts in the IETF (Internet Engineering Task Force) to develop a new class of service models called

Differentiated Services or Diffserv models [2]. The key difference between previously proposed Integrated Services (Intserv) models [3] and Diffserv is that while Intserv provides end-to-end quality of service on a per flow basis, Diffserv is intended to provide long-term service differentiation among the traffic aggregates to different users. In particular, Diffserv pushes the complexity to the network edge, and requires very simple priority scheduling/dropping mechanisms inside the core. While the number of Internet users keeps growing, the Diffserv solution is more suitable because it scales well with increasing number of network users and it does not alter the current Internet paradigm much.

This paper describes an architecture of provisioning guaranteed Internet services. Unlike existing QoS mechanisms such as Intserv and Diffserv, this architecture enables Internet resource management through market forces, e.g., pricing. It is part of a next generation network system currently being developed in the EU funded M3I Project (Market Managed Multi-service Internet)¹. A detailed analysis of two scenarios of the underlying IP network has been presented: ECN capable and Diffserv. We concentrate on the pricing model, its implications for traffic management (e.g., admission control) and possible example services. A simulation framework is also discussed.

2 Guaranteed Service Provider

Here we consider a scenario where a type of guaranteed service is provided to end users that incorporates and extends the classical telephony-like service. Typical applications are those with stringent real-time requirements, such as real-time audio and video services. Their utility functions look like step functions [4], i.e., as soon as the bandwidth share drops below that needed to meet the required delay bounds, the performance falls sharply to zero. Admission control is often necessary for this kind of service.

In M3I, the guaranteed service model consists of two cooperating stakeholders [5]: A stakeholder providing a *basic* communication mechanism and the other making the *refinement* into guaranteed services. At one end, the basic service could be a dynamically priced best-effort service (like current Internet) with no quality guarantees. At the other extreme, no refinement is needed and the basic provider delivers all relevant guarantees directly. In between these extremes, the basic provider can deliver various combinations of price and/or service guarantees. It is envisaged that creating two separate economic entities (the basic Internet service provider (ISP) and the guaranteed service provider (GSP)) would be more convenient for the explicit economic modeling of the provisioning of service guarantees.

The idea of providing guaranteed service over a best-effort IP network infrastructure is based on the work of Gibbens and Kelly [6]. The reference model for a guaranteed service provider is shown in Figure 1. Here we just give a brief introduction of the architecture. For more details (e.g., detailed explanation of

¹ Part of this work was done while the author was on a BT Fellowship at BT Labs, working on the M3I Project.

risk broker, clearing house, etc.), please refer to [5]. The GSP can be viewed as a “layer” similar to that in the OSI model. It is assumed that information streams go through the GSP on its way from the ISP to the customer (via the interfaces, I1, I2, etc.). Inside the GSP domain there are two components: clearinghouse and risk broker. In an environment where the network infrastructure offers no QoS/price guarantees but users nevertheless dynamically adapt to prices or quality signals, a risk broker is needed. It buys dynamically priced communication services with varying qualities from an ISP, and sells transport services with *guarantees* and *simple* end-to-end prices to end customers. In the mean time, the function of a clearinghouse is to gather charges charged to all end customers of some communication, and to redistribute those charges according to some agreement among the end customers.

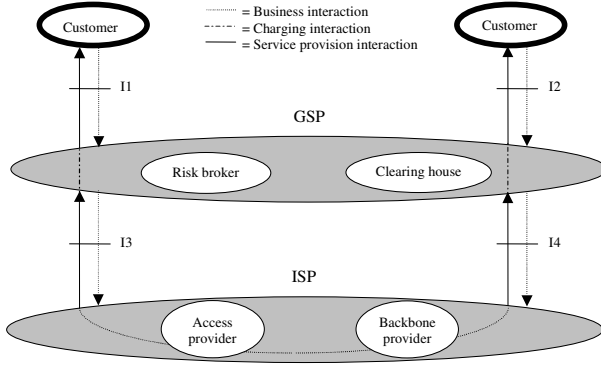


Fig. 1. The GSP reference model (adapted from [5])

There are a range of relations and associated requirements between the ISP and the GSP [5]. Here we consider only two of them.

2.1 Scenario 1

Here the guaranteed service model is similar to that specified in [7]. For traffic conforming to the traffic descriptor and within the agreed maximum duration, the GSP guarantees that packets are delivered between endpoints within a given time limit with a given probability. There are two-level business interactions.

Interface I1/I2 business interactions. The tariffs for the guaranteed service (charged to the end customer) will be of the general form $C = aT + bV + c$, where T is time and V is information volume. Details of this “abc” charging scheme can be found in [8].

Interface I3/I4 business interactions. The ISP offers to the GSP a best-effort datagram service with ECN marking scheme based on congestion pricing

[9]. The tariff to be used for communication is a charge per ECN mark received. Charges apply to the receiver of packets.

2.2 Scenario 2

The difference between this scenario and the previous one is that, the service offered by the ISP to the GSP is Diffserv-like prioritized transfer of datagrams specified by service level agreements (SLAs). Therefore, differences only exist in Interfaces I3/I4. The sender is free to choose priority level on an individual packet basis. SLAs can, however, be dynamically renegotiated. The tariff to be used is a charge per volume in each Diffserv class and the charge per volume increases with priority.

The model of Figure 1 is unique in that the GSP is an economic entity independent of the ISP. In the following sections, we present detailed analyses of the GSP with respect to price interactions, QoS mapping and admission control.

3 The GSP Charging Model

There are N customers (applications) in the system indexed by i . Assume the revenue the GSP receives from customer i is given by

$$C_i = a(F_i, Q_i)T_i + b(F_i, Q_i)V_i + c \quad (1)$$

where F is a set of traffic parameters and Q is a set of quality of service specifications in the traffic contract. T is the duration of the connection and V is the volume. The coefficients a, b, c can be determined from the traffic descriptors as described in [8]. Assume that in Scenario 1 the total number of ECN marks received for connection i is G_i and the price per mark is p_m set by the ISP. Hence the optimisation problem for the GSP is

$$\text{maximise } \sum_i (C_i - G_i p_m) \quad (2)$$

over charges a, b, c , subject to customer's QoS constraints Q_i .

To achieve QoS guarantees the GSP implements call admission control (CAC). Assume that the GSP acts as a CAC gateway by probing the network and accept/reject incoming calls based on ECN marks [6]. If the marking probability P^m is below some threshold A , it accepts a call request. Here A is determined by F_i, Q_i . For example, if Q_i is packet loss probability, then the smaller Q_i is, the smaller A is. So G_i is also a function of F_i, Q_i . On the other hand, smaller A means higher blocking probability P^b , hence smaller number of connections the GSP accepts.

Given a general knowledge of incoming traffic (F_i and Q_i) and ISP's resources (capacity, buffer, etc.), the GSP should set A just small enough to satisfy Q_i so that it can accept more calls. The conjecture is that the revenue from extra customers would exceed the extra cost (marks) incurred². However, if occasionally,

² We later show this conjecture may not be correct using a simple example.

a customer's QoS is violated and the customer's utility decreases, the GSP may have to pay a penalty. The worst consequence would be losing customers. Therefore, the GSP has to maintain a good balance here by means of careful planning and provisioning. This provisioning (among other resource optimisations) should happen on a longer time scale.

3.1 Customer Model

The customer solves the problem

$$\text{maximise } U(F_i, Q_i) - C_i \quad (3)$$

over F_i, Q_i . U is customer's utility function. Here traffic descriptors could be of many different kinds, e.g., a descriptor based on the token bucket.

Although the above GSP model does not seem to be mathematically tractable, it can still give us some insights into such issues as price setting, provisioning and the interaction between customers, the GSP and ISP. We will discuss these issues in the following sections.

4 GSP Provisioning for ECN Charging (Scenario 1)

Given a loss probability (QoS) of P_{loss} and resource capacity of the ISP, the GSP can determine the marking probability P_{mark} under a specific ECN marking scheme, which is normally proportional to P_{loss} . On the basis of this, the GSP can determine the threshold of admission control decisions and hence have estimates of P_{block} and the maximal load³. Under the "abc" charging scheme [8], the per unit time charge is given by

$$C = a(l, m) + b(l, m)M \quad (4)$$

where M is the measured mean rate, l and m are parameters in the effective bandwidth formula.

To remain profitable, the GSP has to ensure that

$$C \geq P_{mark} M p_m \quad (5)$$

where p_m is the price per mark set by the ISP.

As can be seen from above discussions, if a user has quite stringent QoS (say, very small P_{loss}), naturally the charge components a and b must be high due to the property of effective bandwidth. On the other hand, P_{mark} must be small (otherwise the GSP would reject this connection). So it seems quite obvious that (5) will be satisfied. However, this is at the expense of more strict admission control which may turn away many other connection requests. Thus, the GSP's total revenue $\sum_i C_i$ may not be high.

³ It is similar to the case of capacity provisioning using Erlang's formula in traditional telephone networks.

To see the problem more clearly, here we consider a simple example, just illustrating the trade-off the GSP has to make. Assume homogeneous connections and P_{mark} is increasing exponentially with the number of active connections n^4 , i.e., $P_{mark}(n) \approx he^{gn}$, where g, h are small positive numbers. Hence, the maximum number of connections n_{max} the GSP can accept *with profits* satisfies the following equation:

$$nC = he^{gn}nMp_m \quad (6)$$

Since $n_{max} = \frac{1}{g} \log \frac{C}{hMp_m}$, n_{max} increases with C and decreases with g . This also indicates that admission control is indeed necessary at the GSP because ne^{gn} grows faster than nC as n increases. Note that here admission control is not only an engineering decision (to provide satisfactory QoS to users), but also an economic one (to make profit for the GSP). Alternatively, the GSP may want to maximise its net income by solving:

$$\max_n nC - nhe^{gn}Mp_m \quad (7)$$

It is easy to show that the optimal n^* satisfies the following equation:

$$C - he^{gn}Mp_m - n g h e^{gn} Mp_m = 0 \quad (8)$$

Note that $n^* < n_{max}$, as shown in Figure 2. This can be proved as follows. From (8), we have $C = h \exp(gn^*)Mp_m + n^*gh \exp(gn^*)Mp_m$. Then we have $n_{max} = n^* + \frac{1}{g} \log(n^*g + 1) > n^*$. If the GSP wants to have positive net revenue, it should not accept more than n_{max} calls. On the other hand, it may only accept n^* calls to make *maximum* profit. However, this will result in higher blocking rate and its impact on the business relations between the GSP and its customers is an open question.

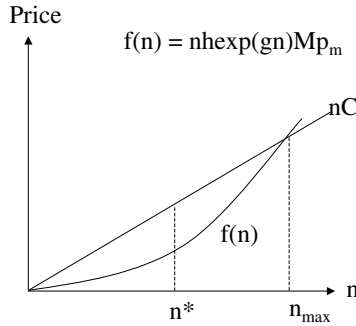


Fig. 2. GSP admission control and charging

⁴ This is a rough assumption derived from the simulation results of [6].

5 GSP Provisioning for the Diffserv Scenario (Scenario 2)

In Diffserv networks, packets are classified into a small number of aggregated flows or “classes”, based on the Diffserv codepoint (DSCP) in the packet’s IP header. This is known as behaviour aggregate (BA) classification. At each Diffserv router, packets are subjected to a “per-hop behaviour” (PHB), which corresponds to the DSCP. The PHB of a behaviour aggregate is distinguished by the DSCP and/or the source/destination address, and source/destination port number. PHBs are implemented at routers by some buffer management and packet scheduling mechanisms. The primary benefit of Diffserv is its scalability, because Diffserv eliminates the need for per-flow state and per-flow processing.

Assume the ISP implements Diffserv with priorities. There are N different classes, with Class 1 has the highest priority. Each priority i is linked with a price per volume, p_i , with $p_1 > p_2 > \dots > p_N$. The ISP advertises to the GSP a set of typical QoS specifications associated with each class (e.g., expected capacity [10]). Then the GSP makes a choice for the priority (and hence forming a service level agreement) on the basis of several factors: customer’s QoS requirements, traffic characteristics, ISP charges, and GSP’s revenue from the customer.

There exists the issue of service mapping. Guaranteed service requests could specify an Intserv service type and a set of quantitative parameters known as a “flowspec”. Requests for guaranteed services must be mapped onto the underlying capabilities of the Diffserv network. Aspects of the mapping include [11]:

- selecting an appropriate PHB, or set of PHBs, for the requested service;
- performing appropriate policing (including shaping or remarking) at the edges of the Diffserv region;
- exporting Intserv parameters from the Diffserv region;
- performing admission control on the Intserv requests that takes into account the resource availability in the Diffserv region.

There is some standard, well-known mapping from Intserv service type to a DSCP that will invoke the appropriate behaviour in the Diffserv network.

Based on this service mapping and the pricing structure of the ISP, the GSP maintains a table (static or dynamic) including the following information: host’s IP address and port number, QoS parameters and traffic characteristics, the minimum priority needed to satisfy QoS, ISP prices for this priority, and the revenue that the GSP gets from the customer. It can be viewed as a contract between the GSP and the ISP. For a static contract, the general structure (contents) of this database may be varied over a long time scale (say, monthly). The GSP can choose the term of the contract, i.e., either long-term fixed contract or short-term flexible contract. In the latter case, the GSP can increase or decrease priorities dynamically. For this kind of dynamic SLA, the GSP functions like a bandwidth broker with a signalling protocol such as RSVP to request for services on demand. For instance, a user normally has a static contract with the GSP for ftp and email applications. The GSP has assigned a low Diffserv priority for this contract, say, Class 3. Now the user wants to use a real-time video service

for a short period and apparently Class 3 is insufficient. So the GSP negotiates with the ISP for a temporary higher priority (say, Class 1) and charges the user more. The pricing policy for this kind of *ad-hoc* service upgrade may be different from the default pricing structure.

To support multiple levels of service, the network provider posts a set of different prices, $p_i, i = 1, 2, \dots, N$. The number of priorities may depend on the population of customers and the desired granularity of resource unit. The ISP also maintains a volume meter for each class for accounting purposes. Similar to telephone networks, the provider could set different prices for these priorities at different times of day, e.g., peak time, evening, weekend, etc. In this case, $p_i^{weekend} < p_i^{evening} < p_i^{peak}$. However, as network traffic is much more bursty and unpredictable than traditional telephone traffic, factors other than time or distance have to be taken into account as well. The detailed pricing policies that the ISP uses may vary from flat rate to usage or congestion-based, or a combination of the above. The ISP also contains a table that indicates the transmit capacity provisioned at each Diffserv service level. This table, in conjunction with the service mapping, is used to perform admission control decisions.

In the above scenario, the actual communication between hosts, the GSP, and the ISP to obtain end-to-end QoS could be based on RSVP.

5.1 Example Service

To support guaranteed delivery or bounded delays in a Diffserv network, it seems that the expedited forwarding (EF) PHB [12] is a natural choice. The EF PHB can be viewed as a virtual leased line service and is implemented as a priority queue that is serviced before all other queues. A number of traffic engineering mechanisms are needed for EF. First, the Diffserv network must be over-provisioned with respect to the EF traffic that it admits. Secondly, admission control must be performed at the edges of the network. Thirdly, edge routers must shape all EF traffic so that their negotiated peak rate is never exceeded. Experiments have shown that EF can provide a low-loss, low-delay and low-jitter end-to-end service.

In this case, the GSP's role is simpler compared to the case of ECN. It is mainly responsible for service mapping. The edge router of the ISP performs CAC. If there is not enough capacity for a new EF request, the GSP can either ask the user to delay his request or downgrade the request to a lower priority (say, best-effort). Since the EF traffic is limited and shaped to a contracted peak rate, it is relatively easy for the ISP to set price. It is also possible for the GSP to work out its own budget and pricing policies. EF traffic would normally be allocated a small percentage of the total network capacity and priced much higher. The GSP can purchase this amount of bandwidth from the ISP at a wholesale price. It will then set (higher) charges a, b, c to sell this EF capacity to multiple customers and make profit.

6 Simulation Experiments

A simulator for guaranteed service provisioning is currently under development within M3I. In this section, we discuss briefly several design considerations of this simulator.

In our simulator, The GSP contains a *QoS agent* that keeps track of the transmit capacity currently available in the router, as well as users' QoS requirements. This information is used to perform admission control. The QoS agent also monitors various QoS metrics and is responsible for QoS re-negotiation. The GSP also has a *pricing agent* which is responsible for metering, accounting, pricing and charging. Here our main focus is on pricing, which involves setting prices (e.g., a, b, c) both dynamically and statically. As discussed before, the pricing agent also interacts with the QoS agent.

At the user side, users try to maximise their net utilities by carefully selecting QoS and traffic parameters. An intelligent agent may be present to estimate user's utility. In the mean time, a user may change his contract with the GSP depending on how he values the quality he receives. The open question is how to determine utility functions.

At the ISP, it may implement ECN or Diffserv. In the ECN scenario, we could consider the virtual queue marking method [6] or the simpler threshold-based marking scheme. The ISP charges the GSP a flat rate fee plus congestion costs based on ECN marks. In the Diffserv scenario, the router maintains a set of queues where each queue represents a behaviour aggregate. Each BA provides a specific QoS. The router will add an arriving packet to a proper BA according to the DSCP (which is determined by the GSP using service mapping). The router meters the traffic in each class and charges by traffic volume accordingly.

7 Conclusion

This paper describes an architecture of provisioning guaranteed Internet services. Unlike existing QoS mechanisms such as Intserv and Diffserv, this architecture enables Internet resource management through market forces, e.g., pricing. We have discussed the GSP model and presented a detailed analysis of two scenarios of the GSP provisioning: ECN capable Internet and Diffserv Internet. We concentrate on the pricing model, its implications for traffic management (e.g., admission control), service quality contract, and possible example services. A simulation framework is also discussed.

Acknowledgements

This work is supported by a BT Fellowship within the M3I Project and the EPSRC grant GR/R18536/01. I would like to thank Dave Songhurst for many helpful discussions.

References

1. Z. Wang. Towards scalable bandwidth allocation in the Internet. Technical report, Bell Labs, Lucent Technologies, 1998.
2. S. Blake et al. An architecture for differentiated services. *Internet RFC 2475*, 1998.
3. R. Braden et al. Integrated services in the Internet architecture: an overview. *Internet RFC 1633*, 1994.
4. S. Shenker. Fundamental design issues for the future Internet. *IEEE JSAC*, 13(7):1176-1188, 1995.
5. The M3I Consortium. Requirements specifications: reference model. Technical report, www.m3i.org/results/main.html, 2000.
6. R. Gibbens and F. Kelly. Distributed connection acceptance control for a connectionless network. In *ITC 16*, 1999.
7. S. Shenker et al. Specification of guaranteed quality of service. *Internet RFC 2212*, 1997.
8. D. Songhurst and F. Kelly. Charging schemes for multi-service networks. In *ITC 15*, pages 781-790, 1997.
9. F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237-252, 1998.
10. D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *IEEE/A CM Trans. Networking*, 6(4):362-373, 1998.
11. Y. Bernet et al. A framework for integrated services operation over Diffserv networks. *Internet Draft*, 2000.
12. V. Jacobson et al. An expedited forwarding PHB. *Internet Draft*, 1998.

Price Optimization of Contents Delivery Systems with Priority

Kyoko Yamori^{1,2}, Yoshiaki Tanaka^{2,1}, and Haruo Akimaru³

¹ Telecommunications Advancement Organization of Japan

² Global Information and Telecommunication Institute, Waseda University

1-21-1 Nishi-Waseda, Shinjuku-ku, Tokyo 169-0051, Japan

kyoko@giti.or.jp, ytanaka@mn.waseda.ac.jp

³ Graduate School of Business Administration Asahi University

Hozumi, Gifu 501-0296, Japan

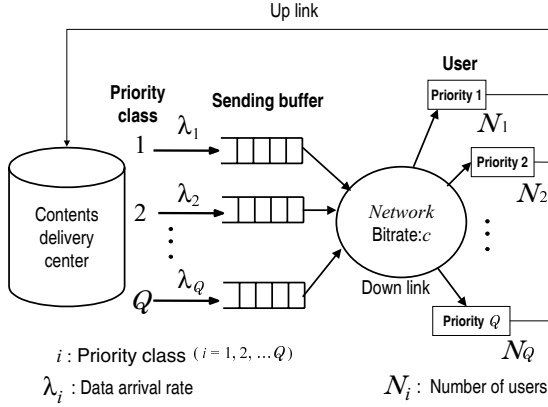
akimaru@alice.asahi-u.ac.jp

Abstract. For data delivery systems such as video-on-demand service, an optimum design is presented to maximize the revenue of the system with priority classes. The willingness-to-pay (WTP) is introduced for a measure of utility (price), and the optimum design is discussed to maximize the revenue. For the system with two priority classes, the optimum condition is given in terms of the traffic load, waiting time for service and pricing for the priority and non-priority classes. In this paper, we use the WTP as the measure of pricing. And we would like to examine the optimum design to propose the optimum pricing methods for time-base and flat-base. In this paper, using the WTP, the utility (price) of the data delivery system has been quantified, and the optimum condition to maximize the revenue of the system has been analyzed. From the numerical examples, the optimum condition for the service grade (waiting time for service) and pricing for priority and non-priority classes has been discussed.

1 Introduction

For data delivery systems such as video-on-demand service, an optimum design is presented to maximize the revenue of the system with priority classes. The willingness-to-pay (WTP) is introduced for a measure of utility (price), and the optimum design is discussed to maximize the revenue. For the system with two priority classes, the optimum condition is given in terms of the traffic load, waiting time for service and pricing for the priority and non-priority classes.

Section 2 describes the traffic model to evaluate the waiting time for service, Section 3 introduces an approximate function for the willingness-to-pay, and Sections 4 and 5 describe the optimum design to propose the optimum pricing methods for time-base and flat-base. Section 6 summarizes the results obtained and gives concluding remarks.

**Fig. 1.** Preemptive priority traffic model

2 Traffic Model

In this paper, we consider a preemptive priority model. Fig.1 shows the model with Q priority classes. Suppose that higher priority class data preempt lower priority class data in service if any, and the preempted remaining data are transmitted afterward. This model is called *preemptive resume model*.

Assume that requests of the data delivery occur at random (Poisson distribution) and that sufficiently large (infinite) buffers are provided. Using the preemptive priority resume model, we investigate the waiting time to start the service for data delivery service. If the data are transmitted immediately upon the request, the requests rate is equivalent to the data arrival rate in Fig.1.

The mean waiting time W_i for priority class i to start the service is given by [1, p.82]

$$W_i = \frac{\sum_{j=1}^i \lambda_j h_j^{(2)}}{2 \left(1 - \sum_{j=1}^{i-1} \rho_j \right) \left(1 - \sum_{j=1}^i \rho_j \right)} \quad (1)$$

where λ_j is the data arrival rate (request rate), h_j the mean net data transmission time (without interruption) and $h_j^{(2)}$ the second moment of the net data transmission time of priority class j , and $\rho_j = \lambda_j h_j$.

If the data transmission time is exponentially distributed, the net transmission time for the remaining data follows the same exponential distribution according to the Markov property [1, p.7]. In this case, we have the relation $h_j^{(2)} = 2h_j^2$.

Although in actual systems, data may be packetized, the packet level operation is neglected in this paper, and data are assumed to be transmitted continuously unless interrupted.

3 Measure of Utility and Pricing

Let us introduce “Willingness-to-Pay: WTP” as a measure of utility [2] for service [3][4]. The WTP is one of the methods of social sciences, and means a price of “How much willing to pay” for a certain service [5][6]. The WTP is used to estimate the utility of service. In this paper, we use the WTP as the measure of pricing.

Denoting by U the WTP (price) and by W the mean waiting time until the service is available, it is assumed that the rate of increase of WTP, dU/U , is proportional to the rate of decrease of the mean waiting time, $-dW/W$.

Then, we have the relation,

$$\frac{dU}{U} = -k \frac{dW}{W} \quad (2)$$

where k is a proportional coefficient. By integration,

$$\int \frac{dU}{U} = -k \int \frac{dW}{W}$$

we have

$$\log U = C - k \log W \quad (3)$$

where C is an integration constant. Setting $D = \exp C$, we have

$$U = DW^{-k}. \quad (4)$$

The parameter k may be statistically estimated by opinion tests [7][8], while D may be determined to balance the revenue and the system cost. The parameter k shows the construction of user’s decision making. Generally, if the parameter k is nearly 1, user’s utility (service satisfaction) depends heavily on the waiting time of data. On the other hand, the value of k is nearly 0, waiting time has no (or little) influence to user’s utility. It is assumed here that the value of k is independent of D in this paper. This means that only the relative value (ratio) of the WTP (price) is relevant.

4 Proposed Pricing Methods

4.1 Time-Base Pricing Method

Let us consider a simple preemptive priority model with two classes, the priority class and the non-priority class. Assuming that the data volume to be delivered is randomly varying with mean H [Mbyte] same for the two classes, the net data transmission time in the network with bitrate c [Mbps] is approximately

exponentially distributed with mean $h = H \times 8/c [\text{sec}]$, and the second moment $h^{(2)} = 2h^2$.

Letting the data arrival rates of the priority and non-priority classes be λ_1 and λ_2 , respectively, the traffic loads in Erlang (occupancy) of the respective classes are given by

$$\rho_1 = \lambda_1 h, \quad \rho_2 = \lambda_2 h. \quad (5)$$

Denote the mean waiting times until the first part of the data is received from the service request, for the priority class and the non-priority class by W_1 and W_2 , respectively. Then, from (1) we have

$$\begin{aligned} W_1 &= \frac{\rho_1}{1 - \rho_1} h \\ W_2 &= \frac{\rho}{(1 - \rho_1)(1 - \rho)} h \end{aligned} \quad (6)$$

where

$$\rho = \rho_1 + \rho_2 \quad (7)$$

is the total traffic load (occupancy) of the network.

Applying (4), the WTP's U_1 and U_2 for the priority and non-priority classes, respectively, are approximated by

$$U_1 = DW_1^{-k}, \quad U_2 = DW_2^{-k}. \quad (8)$$

In the time-base pricing system, set U_1 and U_2 as the priority and non-priority class prices per unit time, respectively. For the priority class, the mean revenue per request is given by

$$U_1 h = \frac{U_1 H}{c} \quad (9)$$

which is proportional to the volume H of the data transmitted. Since λ_1 is the number of requests per unit time for the priority class, its revenue per unit time is given by

$$\lambda_1 U_1 h = \rho_1 U_1. \quad (10)$$

In a similar way, the revenue of non-priority class per unit time is given by

$$\lambda_2 U_2 h = \rho_2 U_2. \quad (11)$$

Hence, we have total revenue per unit time,

$$R = \rho_1 U_1 + \rho_2 U_2. \quad (12)$$

From the relation

$$\frac{\partial R}{\partial \rho} = 0, \quad \frac{\partial R}{\partial \rho_1} = 0 \quad (13)$$

the optimum condition to maximize R in (12) is given by (See Appendix 1.)

$$\frac{1}{\rho - \rho_1} = \frac{k}{\rho(1 - \rho)} \quad (14)$$

$$\left(\frac{1 - k}{\rho_1} - \frac{k}{1 - \rho_1} \right) f_1 = \left(\frac{1}{\rho - \rho_1} + \frac{k}{1 - \rho_1} \right) f_2$$

where

$$f_1 = \rho_1 U_1, \quad f_2 = \rho_2 U_2. \quad (15)$$

4.2 Flat-Base Pricing Method

Denote the number of users of the priority and non-priority classes by N_1 and N_2 , respectively, and the request rates (number of requests per unit time) per user of the respective classes by σ_1 and σ_2 . Assume that σ_1 and σ_2 are constant values not relevant to the waiting time and price in the flat-base pricing. The traffic loads in Erlang (occupancy) of the respective classes, ρ_1 and ρ_2 are given by

$$\rho_1 = N_1 \sigma_1 h, \quad \rho_2 = N_2 \sigma_2 h. \quad (16)$$

Setting priority and non-priority flat-base prices (*e.g.* per month) corresponding to U_1 and U_2 , respectively, the revenue R is given by

$$R = N_1 U_1 + N_2 U_2. \quad (17)$$

If the total number of users N is defined as

$$N = N_1 + N_2 \quad (18)$$

the condition to maximize R is given by

$$\frac{\partial R}{\partial N} = 0, \quad \frac{\partial R}{\partial N_1} = 0. \quad (19)$$

Using the same notation as in the time-base method, from (19) we have the optimum condition, (See Appendix 2.)

$$\frac{1}{\rho - \rho_1} = \frac{1}{\rho(1 - \rho)} \quad (20)$$

$$\left(\frac{1 - k}{\rho_1} - \frac{k}{1 - \rho_1} \right) f_1 = r \left(\frac{1}{\rho - \rho_1} + \frac{k}{1 - \rho_1} \right) f_2$$

where $r = \sigma_1 / \sigma_2$ (constancy value). In the case of $r = 1$, (20) is equivalent to (14).

5 Example Calculations

5.1 Time-Base Pricing Method

Fig.2 shows the graph of R/D for $k = 0.5$. In the case of $k = 0.5$, from (14), we have the optimum condition to maximize R as follows:

The traffic load $\rho = 0.633$ erl ($\rho_1 = 0.168$ erl and $\rho_2 = 0.465$ erl) gives the maximum revenue $R/D = 0.697$.

The ratios of mean waiting time and price (WTP) are:

$$\frac{W_1}{W_2} = \frac{0.202}{2.07} = \frac{1.00}{10.2}, \quad \frac{U_1}{U_2} = \frac{2.23}{0.695} = \frac{3.19}{1.00}.$$

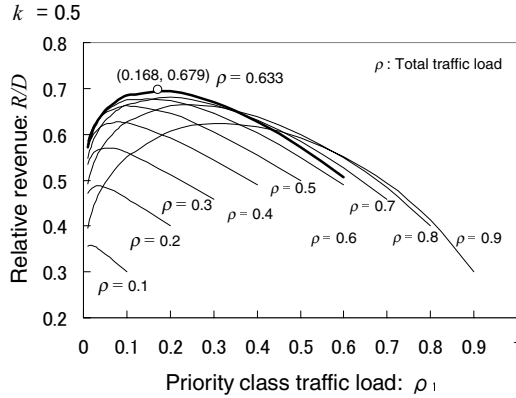


Fig. 2. Graph of R/D

For example, with the data of mean $H = 10$ Mbyte and the network bitrate $c = 8$ Mbps, we have $h = H \times 8/c = 10$ sec. Hence, the mean waiting time for starting service are $W_1 = 2.02$ sec and $W_2 = 20.7$ sec, respectively, for the priority and non-priority classes. If the price for the non-priority class is set at 100 Yen per 10 Mbyte data, the corresponding price for the priority class becomes 319 Yen, according to the ratio of the price (WTP).

5.2 Flat-Base Pricing Method

In the case of $k = 0.5$, Table 1 shows the optimum condition for $r = 0.5$, 1.0 and 2.0 calculated from (20). The case of $r = 1.0$ is the same as in the time-base pricing.

Table 1. Optimum Condition

r	ρ	ρ_1	R/D	W_1	W_2	U_1	U_2
0.5	0.7380	0.3513	0.5702	0.5416	4.3427	1.3588	0.4799
0.1	0.6325	0.1677	0.6968	0.2015	2.0683	2.2279	0.6953
2.0	0.5391	0.0421	1.1003	0.0440	1.2209	4.7688	0.9050

- (1) Case of $r = 0.5$: Priority class request is a half of the non-priority request. The ratios of mean waiting time and price (WTP) are:

$$\frac{W_1}{W_2} = \frac{0.542}{4.34} = \frac{1.00}{8.02}, \quad \frac{U_1}{U_2} = \frac{1.36}{0.480} = \frac{2.83}{1.00}.$$

When $h = 10$ sec and $\sigma_1 = 1.0/\text{hr}$, then $\sigma_2 = 2.0/\text{hr}$, the optimum numbers of users are given by

$$N_1 = \frac{\rho_1}{\sigma_1 h} = 0.351 \times \frac{3,600}{1.0 \times 10} = 126.5 \rightarrow 126$$

$$N_2 = \frac{\rho - \rho_1}{\sigma_2 h} = 0.387 \times \frac{3,600}{2.0 \times 10} = 69.60 \rightarrow 69.$$

- (2) Case of $r = 2.0$: Priority class request is twice of the non-priority request. The ratios of mean waiting time and price (WTP) are:

$$\frac{W_1}{W_2} = \frac{0.044}{1.22} = \frac{1.00}{22.8}, \quad \frac{U_1}{U_2} = \frac{4.77}{0.901} = \frac{5.27}{1.00}$$

When $h = 10$ sec and $\sigma_1 = 1.0/\text{hr}$, then $\sigma_2 = 0.5/\text{hr}$, the optimum numbers of users are given by

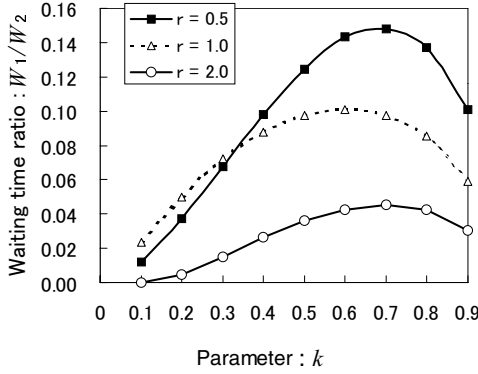
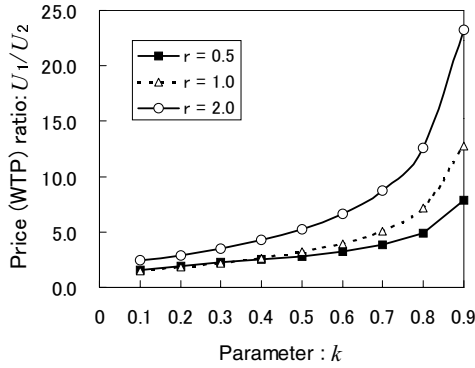
$$N_1 = \frac{\rho_1}{\sigma_1 h} = 0.0421 \times \frac{3,600}{1.0 \times 10} = 15.16 \rightarrow 15$$

$$N_2 = \frac{\rho - \rho_1}{\sigma_2 h} = 0.497 \times \frac{3,600}{0.5 \times 10} = 357.8 \rightarrow 357.$$

Figs.3 and 4 show the effect of the value of k and r . Fig.3 indicates the optimum ratio, W_1/W_2 , of waiting time for service for priority and non-priority classes, which has a hump around $k = 0.7$. Fig.4 shows the optimum ratio of price (WTP), U_1/U_2 .

6 Conclusions

In this paper, using the WTP (Willingness-to-Pay), the utility (price) of the data delivery system has been quantified, and the optimum condition to maximize the revenue of the system has been analyzed. From the numerical examples, the

**Fig. 3.** Optimum ratios of waiting time**Fig. 4.** Optimum ratios of price (WTP)

optimum condition for the service grade (waiting time for service) and pricing for priority and non-priority classes has been discussed.

The assumptions and parameters for the WTP formula (4) need to be verified statistically. As mentioned in Section 3, it is assumed that the value of k is not relevant to the value of D in the utility measure function. This point should be clarified by the opinion tests, etc. Even if more sophisticated formulas for the WTP are used, the framework of this paper may be applied in a similar manner. Although the request rates per user are assumed constant in the flat-base pricing, they might be influenced by the price and waiting time, for which model the analyses are also under study.

A fixed value of network bitrate c and exponential service time are assumed in (6) for estimating the mean waiting time for service. The assumption may be applicable for a network with the bandwidth reservation scheme such as the CBR

(constant bit rate) in the ATM (asynchronous transfer mode), and Diffserve in the Internet. Even for the case of the bitrate varying randomly with mean c , the results of this paper may be applicable as far as the data transmission time is approximately exponentially distributed.

References

1. Akimaru, H. and Kawashima, K.: Teletraffic - Theory and Applications, Second Edition, Springer-Verlag (1999)
2. Keeney, R.L. and Raifa, H.: Decision with Multiple Objectives, Preferences and Value Tradeoffs, John Wiley & Sons (1976)
3. Mitchell, R.C. and Carson, R.T.: Using Surveys to Value Public Goods, The Contingent Valuation method, Resources for Future (1989)
4. Hausman, J.A.: Contingent Valuation - A Critical Assessment, North-Holland (1993)
5. Bjornstad, D.J. and Kahn, J.R.: The Contingent Valuation of Environmental Resources, Edward Elgar (1996)
6. Hayashi, Y.: Evaluation of the Benefits for Environmental Amenities, Other Non-market Goods and Services by the Contingent Valuation Method, JSCE, Vol. 83 (1998) 37–40
7. Akimaru, H., Yamori, K. and Finley, M.R.: Dimensioning of Video Conferencing Systems in ATM Networks, ICC'99, T11-2, Tokyo (1999) 294–299
8. Yamori, K., Akimaru, H. and Finley, M.R.: Optimum Design of Videoconferencing Systems over ATM Networks, ICCT 2000, S05, Beijing (2000) 1213–1219

Appendix 1: Derivation of Equation (14)

From the second equation in (13),

$$\frac{\partial R}{\partial \rho_1} = f_1 \frac{\partial}{\partial \rho_1} \log f_1 + f_2 \frac{\partial}{\partial \rho_1} \log f_2 = 0. \quad (21)$$

Since $f_1 = \rho_1 U_1 = D(1 - \rho_1)^k \rho_1^{1-k}$, taking the logarithm we have

$$\log f_1 = \log D + k \log(1 - \rho_1) + (1 - k) \log \rho_1. \quad (22)$$

Differentiating (22) w.r.t. (with respect to) ρ_1 yields

$$\frac{\partial}{\partial \rho_1} \log f_1 = \frac{1 - k}{\rho_1} + \frac{k}{1 - \rho_1}. \quad (23)$$

In a similar way,

$$\begin{aligned} \log f_2 &= \log D + k \log(1 - \rho_1) + k \log(1 - \rho) \\ &\quad + \log(\rho - \rho_1) - k \log \rho, \end{aligned} \quad (24)$$

Differentiating (24) w.r.t ρ_2 , we have

$$\frac{\partial}{\partial \rho_1} \log f_2 = -\frac{k}{1 - \rho_1} - \frac{1}{\rho - \rho_1}. \quad (25)$$

Substituting (23) and (25) in (21), the second equation in (14) follows.

Noting that $\partial f_1 / \partial \rho = 0$ since W_1 in (6) includes no ρ , from the first equation in (13) we obtain

$$\frac{\partial R}{\partial \rho} = f_2 \frac{\partial}{\partial \rho} \log f_2 = f_2 \left(\frac{1}{\rho - \rho_1} - \frac{k}{\rho(1 - \rho)} \right) = 0. \quad (26)$$

Assuming $f_2 \neq 0$ we get the first equation in (14)

For numerical computation, solving the first equation in (14) w.r.t. $\rho > 0$, we have

$$\rho = \frac{1 - k + \sqrt{(1 - k)^2 + 4k\rho_1}}{2}. \quad (27)$$

Using (27) in the second equation in (14), we can compute the optimum solutions for ρ and ρ_1 by iteration.

Appendix 2: Derivation of Equation (20)

From (16) and (17), the revenue R is expressed as

$$R = \frac{\rho_1}{\sigma_1} U_1 + \frac{\rho_2}{\sigma_2} U_2. \quad (28)$$

Define

$$f = f_1 + r f_2 \quad (29)$$

where $f_1 = \rho_1 U_1$, $f_2 = \rho_2 U_2$ and $r = \sigma_1 / \sigma_2$. Then, using (29) in (28), we have

$$R = \frac{f}{\sigma_1}. \quad (30)$$

Applying the differentiation formula for a function of a function, the optimum condition (19) becomes

$$\frac{\partial R}{\partial N} = \frac{\partial R}{\partial \rho} \frac{\partial \rho}{\partial N} = 0, \quad \frac{\partial R}{\partial N_1} = \frac{\partial R}{\partial \rho_1} \frac{\partial \rho_1}{\partial N_1} = 0. \quad (31)$$

Noting that the total occupancy ρ is expressed as

$$\rho = \rho_1 + \rho_2 = N_1 \sigma_1 + (N - N_1) \sigma_2 = N_1(\sigma_1 - \sigma_2) + N \sigma_2, \quad (32)$$

and σ_1 and σ_2 are constant values, we have

$$\frac{\partial R}{\partial \rho} = \frac{1}{\sigma_2} \frac{\partial f}{\partial \rho}, \quad \frac{\partial R}{\partial \rho_1} = \frac{1}{\sigma_1} \frac{\partial f}{\partial \rho_1} \quad (33)$$

$$\frac{\partial \rho}{\partial N} = \sigma_2, \quad \frac{\partial \rho_1}{\partial N_1} = \sigma_1. \quad (34)$$

Using (33) and (34) in (31), we have

$$\frac{\partial f}{\partial \rho} = \frac{\partial f_1}{\partial \rho} = 0, \quad \frac{\partial f}{\partial \rho_1} = \frac{\partial f_1}{\partial \rho} + r \frac{\partial f_2}{\partial \rho_1} = 0. \quad (35)$$

In a similar manner as in Appendix 1, from (35) the optimum condition (20) is derived for the flat-base pricing method.

An Approach to Internet-Based Virtual Call Center Implementation

M. Popovic and V. Kovacevic

University of Novi Sad, Fruskogorska 11, 21000 Novi Sad, Yugoslavia
Tel: +381-21-350-701, Fax: +381-21-350-610,
{rt_pop, rt_kovac}@krt.neobee.net

Abstract. The era of classical PBX-based call centers has passed. Those systems were proprietary and closed, i.e. with fixed functionality. Today, the Internet and multimedia applications are becoming more and more popular across the world, and there is a lot of effort in both academia and industry to build and deploy modern Internet-based call centers. This paper should be viewed as a contribution to these efforts. It presents our approach to Internet-based virtual call center implementation. In contrast to other efforts, we consider the virtual call center as a universal infrastructure, which could be used as a telecommunication management network center and as an intelligent network service control point, too. In the paper we present our concept, the most interesting implementation details, and pilot network configuration.

1. Introduction

The paper presents an approach to Internet-based Virtual Call Center (VCC) implementation. The approach is based on the experience gained from research and development of classical call centers [1-4] and similar systems for Russian telecommunication network [5-10]. The main lessons learned from this effort are the following:

- End customers as well as software developers fully depend on hardware equipment manufacturer.
- A classical call center consists of two loosely coupled parts: PBX-based (Private Branch Exchange) ACD (Automatic Call Distributor) and LAN (Local Area Network), which operate almost independently.
- A classical call center serves PSTN (Public Switched Telephone Network) users only. There is no possibility to access the call center from other networks.
- The quality of subscriber-agent communication is pure because it is based on voice only.
- Proprietary ACD is a closed system, and it is hard to extend its functionality and/or capacity.

In order to overcome these problems, the call center must be Internet/Intranet based, and it has to be decentralized, i.e. virtual. This Virtual Call Center (VCC) should provide services to all kinds of users, including telephone network subscribers and Internet subscribers.

The core of the Virtual Call Center is the Intranet, which is based on TCP/IP technology. Physical infrastructure could be Ethernet for smaller VCCs, and FDDI (Fiber Distributed Data Interconnect) or ATM (Asynchronous Transfer Mode) for bigger VCCs. The VCC Intranet is connected to the global Internet through the Internet firewall, and to PSTN through PSTN gateway.

In our approach, the VCC is the infrastructure, which could be used as TMN (Telecommunication Management Network) center, and/or SCP (Service Control Point) of IN (Intelligent Network). We plan to keep the system open so that new IN services (not yet standardized by the ITU-T) could be introduced in the future, too.

The text of the paper is organized as follows. Next subsection briefly cites the related work relevant to the paper. Section 2 presents our concept of the Internet-based virtual call center. In Section 3 we describe the main implementation details. In Section 4 we describe the pilot network, which is used for the experimental evaluation of our approach. Section 5 contains the conclusions. At the end of the paper we give the list of the references cited throughout the paper.

1.1 Related Work

Our approach is one of the efforts to improve existing call centers, which are undertaken in both academia [10-15] and industry. For example, The Cisco Systems is promoting its Architecture for Voice, Video and Integrated Data (AVVID) [16]. The 3Com have classified the software-based call centers into Virtual Call Centers, Web-enabled Call Centers, and Multimedia Call Centers [17]. The Dialogic advocates Internet-enabled Call Centers [18]. However, in our vision the VCC will be used for servicing both users/subscribers and other networks. For example, we plan to use a VCC as a platform for TMN, too. Additionally, VCC could add new functionality to the existing networks. For example, VCC will provide IN platform for the existing PSTN.

2. Internet-Based Virtual Call Center Concept

A classical call center, which consists of PBX (Private Branch eXchange) and computer workstations and servers connected to the LAN (Local Area Network), is becoming obsolete today. The main reason for this situation is that the classical call center is a proprietary closed system with fixed functionality. The quality of voice-based communication between a customer and an agent is poor. In order to eliminate these disadvantages, a modern call center must be Internet/intranet based, and it has to be decentralized, i.e. virtual. Such virtual call center (VCC) should provide services to all kinds of users, including the following user categories:

- Analog subscribers. The call is made from the plain old telephone, connected to the PSTN.
- Digital subscribers. The call is made from the ISDN (Integrated Services Digital Network) terminal, which is connected to the PSTN/GSTN (Global Switched Telephone Network) through BRI/PRI (Basic Rate Interface / Primary Rate Interface).

- Internet subscribers using a browser to access the call center.
- Internet/Intranet subscribers using H.323 terminal to access the services offered by the call center.
- Internet E-Commerce Transaction users/subscribers and other database oriented services.

The structure of the VCC is shown in Figure 1. The core of Virtual Call Center is an Intranet, which is based on TCP/IP technology. Physical infrastructure could be Ethernet for smaller VCCs, and FDDI (Fiber Distributed Data Interconnect) or ATM (Asynchronous Transfer Mode) for bigger VCCs.

The AOM (Administration, Operation & Maintenance) functions should be provided through SNMP (Simple Network Management Protocol), which is the most mature technology comparing to other similar technologies. However, VCC could offer TMN (Telecommunication Management Network) services to PSTN operator through SS7 (Signaling System number 7) platform provided by the VCC.

VCC services will be provided by different kinds of Agents, including the following categories:

- Simple telephone-call agent. This agent has just a plain old telephone, and he/she just answers the telephone calls distributed by the CTI Server.
- Call Agent with database support. This agent uses H.323 terminal (a PC with a LAN card, sound blaster, and hands-free headset), and he/she can issue database queries while servicing a user/subscriber. H.323 conferencing with a user/subscriber and/or other agent(s) is possible, too.
- Call & E-mail Agent. This agent uses H.323 terminal to answer voice calls and to replay user/subscriber e-mail messages (which could include text, audio and video).
- Other kinds of Agents could be added later. For example, H.323 remote agent working at home, remote agent with the Bluetooth headset, etc.

VCC services are provided using different kinds of servers, including the following categories:

- DNS (Domain Name Server) & Proxy server.
- Mail server.
- WWW (World Wide Web) server.
- H.323 Gatekeeper (H.323 address translation and administration).
- H.323 MCU used for H.323 multimedia conferencing.
- AOM (Administration, Operation & Maintenance) server.
- CTI (Computer Telephony Integration) Server. It distributes calls, maintains the queues for individual services/agents, and supports CTI services inside the VCC. There could be more than one CTI server in the system, which operate in the load-sharing working mode.
- IVR (Interactive Voice Response) Servers.
- Application server(s), for different kinds of tasks.
- Database server(s).

The VCC is connected to Internet through the Bastion host based Firewall. In future we plan to introduce VPN (Virtual Private Networking) switch, too.

VCC is connected to PSTN through different kinds of Gateways, including the following categories:

- **Analog VoIP Gateway.** This gateway is connected to the PSTN through the series of plain old analog-subscriber lines. It is seen by PSTN as a virtual PBX. The lines are so-called PBX series lines. This means that LE (Local Exchange) will place a new call to the PBX on the first available line from this series. This is a traditional way of connecting a PBX to PSTN and it is the most effective solution for the small VCCs. Analog Gateway is a PC with a LAN card, and a series of commercial-of-the-shelf modems and/or media cards. The voice transmission over Intranet should be with minimal compression in order to provide high-quality voice connections. For example, G.711 could be used for calls coming from PSTN. Additionally the Intranet should use traffic classification (Layer 3 schemes such as IP precedence or use of the Different Service Code Point, Layer 2 schemes such as 802.1P and also use of Real-Time Protocol) and traffic prioritization technology, in order to guarantee an acceptable QoS for user/subscriber-agent voice connections.
- **Digital VoIP Gateway.** This gateway is connected to PSTN through E1/DSS1, interface (30B+D/DSS1 Terminal side). The gateway is a PC with a LAN card and an E1/HDLC/Speech card. The card could be commercial-of-the-shelf or proprietary.
- **IN (Intelligent Network) Gateway.** This gateway connects to TE (Transit Exchange) in PSTN over 2 or more (in future) E1/SS7 interfaces. The gateway is a PC with a LAN card and E1/HDLC/Speech SS7 card(s).
- Some other kinds of Gateways could be added later.

The Gateways in our terminology are also known as “front-end” computers. The solution is highly scalable. For small VCCs one front-end computer suffices. For larger ones, for front-ends could be added. This solution takes care about reliability issues, too. For reliability purposes, backup (reserve) front-ends could be added.

In case when more than one PSTN gateway exists in the system, they operate in the load-sharing working mode. For the incoming calls to VCC, PSTN will view the set of PSTN gateways as a group of trunks, and it will allocate the trunk (channel) to be used for the connection. On the other hand, in the case of the outgoing call from VCC, CTI server that handles the call will offer the call to all available PSTN gateways. The PSTN gateway, which replays first, will continue to handle the call. The requests to other PSTN gateways will be postponed.

Obviously for a system such as VCC the security of the system is the crucial issue. Information contained in, and the services offered by VCC must be protected. Resistance to attacks from PSTN is provisioned through correct implementation of the applied signaling system (for example DSS1). Database inquiries/transactions are secured by PINs (Person Identification Numbers). The attacks from Internet are prevented by firewall, and VPN support. The providers using the same VCC infrastructure are separated by the VLANs (Virtual LANs) support.

Another important issue is the extendibility of the system. The system is designed in such a way that the introduction of new services doesn't affect the agent call handling, i.e. how he accepts, transfers, or releases the call. Furthermore, the system supports CCR (Customer Controlled Routing) functionality through the CCR scripts. The CCR support enables call-processing software reusability.

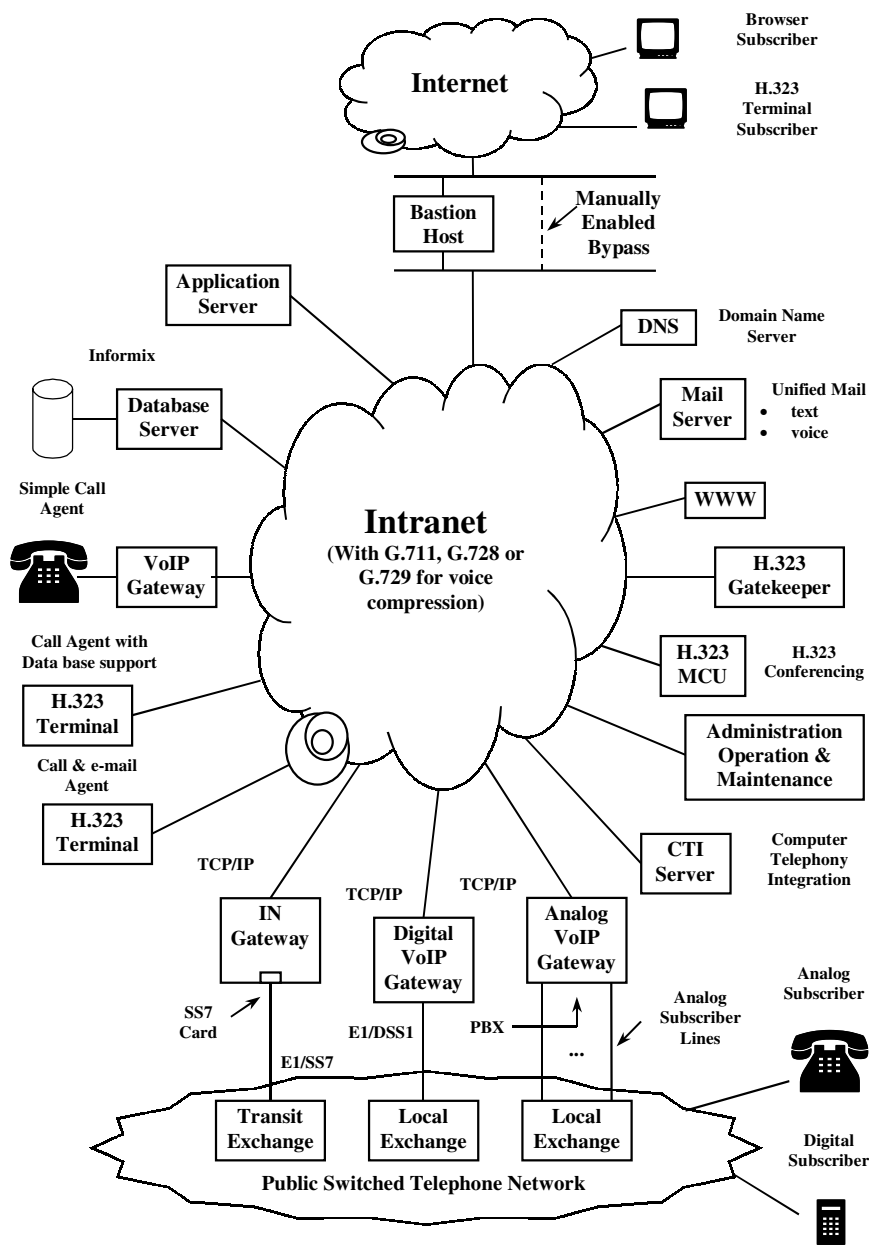


Fig. 1. The Virtual Call Center Structure

Of course, if a new service is introduced the corresponding application and database server programs must be deployed. To make it easier to introduce a new service we are using component-based software design and Commercial-of-the-Shelf (COTS) components. Additionally, instead of simple TCP/IP client-server architectural model, we use CORBA-based three-tier client—application-server—database-server architectural model.

Let us summarize this section of the paper. VCC can distribute calls coming from users of PSTN and/or Internet to the agents. Moreover, certain computers in VCC could act as a SCP, which is connected, to one or more SSPs (Service Switching Points). These dedicated computers automatically perform the SCP functions. Of course, VCC personal does SCP administration, operation and maintenance. Finally, VCC could be used as TMN center, too. In that case VCC agents act as TMN operators, who supervise and control the telecommunication network.

3. Internet-Based Virtual Call Center Implementation

As the most interesting implementation details we have selected the following:

- Local voice call (i.e. virtual PBX call) processing.
- PSTN inbound (incoming) call processing.
- Internet inbound call processing.

The components involved in a local call processing are the following:

- According to ITU-T Q.71: FE1, FE2, FE4, and FE5
- AB analysis (ABA) component. This component analyses calling and called party numbers.
- IVR (Interactive Voice Response) component. This component plays RANs (Recorded Announcements) to a user/subscriber, and accepts his selections.
- Call Distributor (CD), Service working Group (SG), and AGENT components.

Static component relations are shown in Figure 2. The standard Q.71 FE1-FE2-FE4-FE5 structure has been extended with ABA for B analysis, CD, SG, and AG for ACD functionality and IVR for interactive voice response functionality.

Dynamic component relations are shown in Figures 3, in the form of MSC (Message Sequence Charts). Figure 3 shows the telephony-related component interaction.

Standard Q.71 MSC has been customized for the ACD (Automatic Call Distribution) functionality. The message inserts are as follows:

- CD signals a new call arrival to SG with the “NEW_CALL” message.
- SG accepts a call with the “SG_CALL_ACCEPTED” message.
- A welcome message and the music is played with the sequence of “PLAY” messages.
- SG offers a call to an agent with the “CALL_OFFER” message.
- AG accepts a call with the “CALL_ACCEPTED” message.
- SG signals that an agent has accepted the call with the “OPERATOR_READY” message.

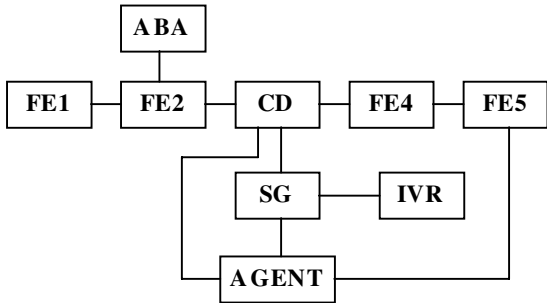


Fig. 2. The static component relations for a local call processing

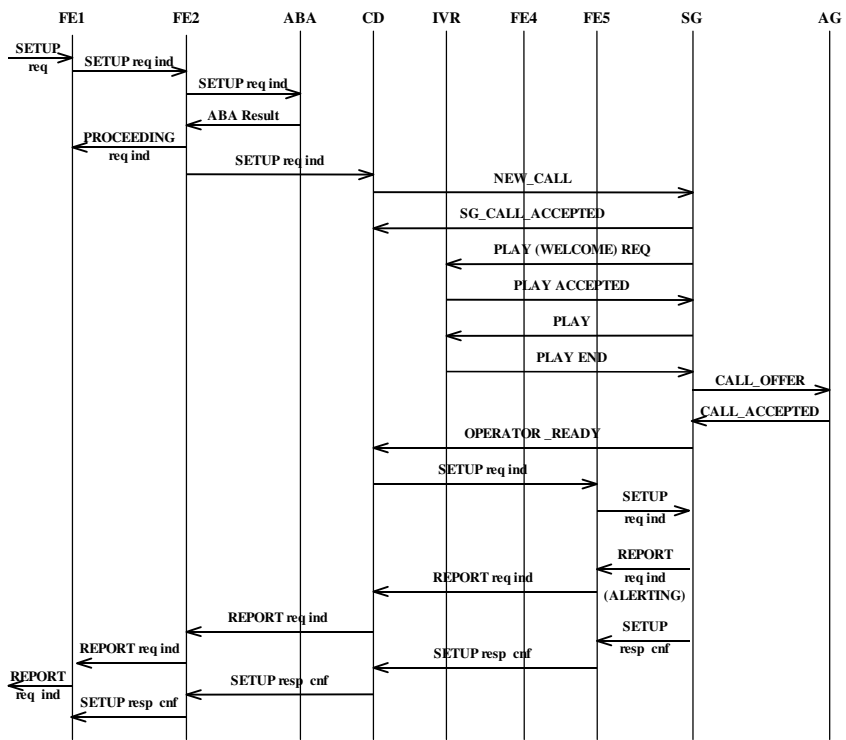


Fig. 3. The telephony-related MSC for a local call processing

The most important aspect of the Internet-related component interaction is the usage of IP multicasting. IP multicasting is used for two purposes. The first one is for playing an announcement and/or music to the calling party, as follows:

- “PLAY_REQ” (play request) message is sent, using IP multicasting, to all active IVRs, which are registered in the corresponding multicast group. It should be

noticed that at least one IVR must be active, but there is a possibility to have more of them in the system.

- IVRs replay with the “PLAY ACCEPTED” message, using IP unicast.
- The IVR, which replied first, is going to service a call. It will receive the “PLAY” message. Other IVRs will receive the “DISCONNECT” message.

The second usage of IP multicasting is the following. When a call is released, SG that is involved in its processing should advertise the availability of the agent, which serviced the call.

One more aspect of the Internet-related component interaction has to be clarified. That aspect addresses information (voice) connection establishing and release (open and close). This phase of call processing is handled by H.245 protocol. When a new call arrives into the call center, it comes to CTI server. CTI server together with IVR and selected H.323 agent will process the signaling phase of the call up to the point in which the CTI server sends “REPORT” message to the H.323 client (i.e. new call from PSTN). After that point H.323 client and agent should solely open information channel using H.245 protocol. Involvement of neither CTI server nor IVR is needed in order to establish the information channel between the PSTN subscriber and call center agent.

The components involved in a inbound call processing are the following:

- Q.71: FE6, FE4, and FE5
- AB analysis (ABA)
- IVR
- CD, SG, and AG.

Static component relations are shown in Figure 4. Standard Q.71 FE6-FE4-FE5 structure has been extended with ABA for B analysis, CD, SG, and AG for ACD functionality and IVR for interactive voice recognition.

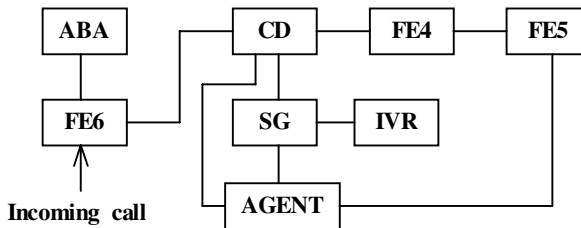


Fig. 4. The static component relations for a PSTN inbound call processing

Dynamic component relations are very similar to those already shown in Figure 3: FE6 plays a role similar to FE2 in local call processing. The difference is the additional interaction of FE6 with the PSTN gateway.

In addition to calls coming from the PSTN, the VCC supports the calls coming from Internet. A user using www browser (“Internet Explorer”, “Netscape Navigator”, or some other) loads the VCC home “.html” page from the VCC www server. This page loads another pages and associated applets in response to user selections. The applets communicate with application server’s CORBA (JAGUAR) components, which in turn communicate with a database server. This is a so-called 3-

tyre system. A special VCC client does the database administration. A user may contact a VCC agent if and when needed by using the “talk-to-me” button.

4. The Pilot Network

For the purpose of experimental evaluation of our approach to VCC implementation the 100Mb/s pilot network with the following configuration will be used:

- Two layer 3 (L3) switches, working back-to-back through 1Gb/s interconnection, using VRRP (Virtual Router Redundancy Protocol) protocol.
- Two layer 2.5 (L2.5) switches, which support two traffic priority queues (high priority and low priority).
- Two simple layer 2 (L2) switches.
- Farm of about 10 servers (CTI, IVR, www, application, database, etc.)
- 50 workstations.
- Internet connection through the router and VPN switch.
- One PSTN gateway.
- One RAS (Remote Access Server).
- One access point for wireless workstations.

For a system such as VCC, the traffic classification and prioritization is the crucial issue. Of course the policy making must be centralized. It is done through Directory Service (DS), which enables the administrative personal to specify priorities for individual applications running in a VCC. Once the “policy” is made it is distributed to individual VCC’s switches. The switches in turn will start operating according to the central VCC’s policy. In this way the call-processing software which we develop will be generic, i.e. independent of the concrete VCC configuration. This is important, because with such a strategy the performance issues are becoming the issues for Intranet dimensioning and traffic policy making. Even on modest configurations VoIP can work with acceptable performance. Of course, IPv6 will solve all of the problems in the future.

Apart from performance evaluation, we expect the pilot network to help us evaluate reliability, scalability, and extendibility issues, too.

5. Conclusions

In this paper we have presented our approach to Internet-based virtual call center implementation. We have described the concept, the most interesting implementation details, and the pilot network configuration.

In addition to other efforts in the filed we view the VCC as a universal infrastructure, which could be used as a TMN center and IN SCP.

References

1. K. Dawson, *Call Center Handbook: The Complete Guide to Starting, Running and Improving Your Call Center*, CMP Books, ISBN: 1578200474, 1999.
2. B. Cleveland and J. Mayben, *Call Center Management on Fast Forward: Succeeding in Today's Dynamic Environment*, Call Center Pr, ISBN: 0965909301, 1999.
3. M. Bodin, K. Dawson, *The Call Center Dictionary: The Complete Guide to Call Center and Help Desk Technology and Operations*, CMP Books, ISBN: 1578200431, 1999.
4. R. A. Gable, *Inbound Call Centers: Design, Implementation, and Management*, Artech House, ISBN: 089006640X, 1993.
5. M. Popovic, V. Kovacevic, S. Svircevic, "Concept of Developed Medium Size Switching System CVHN-RT", *Proceedings of 15th International Teletraffic Congress*, Saint Petersburg, Russia, 1998.
6. V. Kovacevic, M. Popovic, E. Secerov, Z. Manojlovic, and M. Skrbic, "Software Concept Applied in Subscriber Digital Concentrator ACK 2000 for Russian Telecommunication Network", *Proceedings of IEEE International Conference on Telecommunications ICT '98*, Porto Carras, Greece, June 21-25, 1998, Vol. IV, pp. 212-215.
7. M. Popovic, V. Kovacevic, and M. Skrbic, "On Possible V5.2 Concentrator Architectures", *Proceedings of IEEE International Conference on Telecommunications ICT'99*, Cheju Korea, Jun 15-18, 1999.
8. M. Popovic, V. Kovacevic, and M. Skrbic, "Software Development & Testing Methodology used for Subscriber Digital Concentrator ACK-2000", *Proceedings of IEEE 10th International Symposium on Software Reliability Engineering*, Boca Raton, Florida, USA, November 1-4, 1999, pp. 175-178.
9. M. Popovic, V. Kovacevic, and M. Skrbic, "Software Reliability & Maintenance Concept used for Automatic Call Distributor MEDIO ACD", *Proceedings of IEEE 11th International Symposium on Software Reliability Engineering*, San Jose, California, USA, October 8-11, 2000, pp. 336-342.
10. D. Yannakopoulos, M. Ferreti, M. Schultz, "Object Lessons Learned from an Intelligent Agents Framework for Telephony-Based Applications", *IEEE Proceedings of Technology of Object-Oriented Languages and Systems*, Santa Barbara, California, 1-5 August 1999.
11. D. Jevtic and D. Sablic, "Intelligent Call Transfer Based on Reinforcement Learning", *IEEE Proceedings of IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como, Italy, 24-27 July, 2000.
12. H. El-Gendy, "Advanced Wired Follow Me Telephony Service", *IEEE Proceedings of 12th International Conference on Computer Communications*, Seoul, South Korea, August 21-24 1995.
13. R. Nelson, "Technology Tools for the Year 2005: Supporting the Nursing Process", *IEEE Proceedings of Pacific Medical Technology Symposium*, Honolulu, Hawaii, August 17-20, 1998.
14. A. Rohm, G. Pernul, "COPS: A Model and Infrastructure for Secure and Fair Electronic Markets", *IEEE Proceedings of Thirty-second Annual Hawaii International Conference on System Science*, Maui, Hawaii, 5-8 January 1999.
15. Qian-Ping Gu and S. Peng, "Node-to-Node Cluster Fault Tolerant Routing in Hypercubes", *IEEE Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks*, Taipei, Taiwan, December 18-20 1997.
16. "Architecture for Voice, Video and Integrated Data" – White Paper, www.cisco.com
17. "The Call Center Revolution" – Technical Paper, www.3com.com
18. "IP Telephony in the Call Center" – White Paper, www.dialogic.com

Implementation and Characterization of an Advanced Scheduler

Fulvio Risso

Dipartimento di Automatica e Informatica, Politecnico di Torino,
Corso Duca degli Abruzzi, 24, 10129 Torino (Italy)
risso@polito.it

Abstract. Decoupled-CBQ, a CBQ derived scheduler, has been proved being a substantial improvement over CBQ. D-CBQ main advantages are a new set of rules for distributing excess bandwidth and the ability to guarantee bandwidth and delay in a separate way, whence the name "decoupled". This paper aims at the characterization of D-CBQ by means of an extended set of simulations and a real implementation into the ALTQ framework.

1. Introduction

Class Based Queuing [1], (CBQ) and Hierarchical Fair Service Curve [2] (H-FSC) represent an interesting solution to integrated networks that aim to provide hierarchical link-sharing, tighter delay bounds and bandwidth guarantees. However the configuration of HFSC is less intuitive from an Internet Service Provider point of view, therefore CBQ is the most appealing advanced scheduler available today.

An in-depth analysis of CBQ, on the other hand, shows several problems; most of them have been pointed out in [3] and [7]. This paper aims at the completion of [7] by summarizing the D-CBQ characteristics, presenting the implementation issues, and characterizing this new scheduler.

This paper is structured as follows. Section 2 summarizes D-CBQ characteristics; Section 3 presents the simulations used to validate the prototype, while Section 4 discusses the efforts in implementing D-CBQ in a real router. Finally, Section 5 presents some conclusive remarks.

2. Decoupled Class Based Queuing

The most important D-CBQ characteristics include new link-sharing guidelines, the decoupling between bandwidth and delay and the excellent precision in respecting bandwidth and delay.

2.1 New Link-Sharing Guidelines

New link-sharing guidelines require the definition of the new concept of the Bounded Branch Subtree (BBS). All the bounded¹ classes plus all the classes that are child of the root class are called BBS-root. Each BBS-root generates a Bounded Branch Subtree that includes the set of classes that *share* a BBS-root as common ancestor plus the BBS-root itself. BBSs can be embedded; an example can be seen in Fig. 1

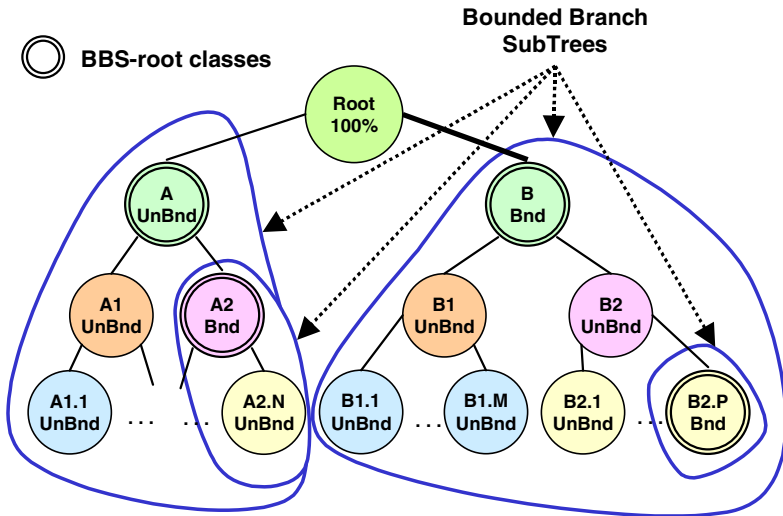


Fig. 1. Bounded Branch Subtrees. Classes can be either unbounded ("UnBnd") or bounded ("Bnd")

Each BBS acts as a new link-sharing hierarchy that is almost independent from the others. A class may belong to several BBSs, therefore it can have several BBS-root classes. Among these BBS-roots, the one with the lowest level in the link-sharing hierarchy is called L-BBS-root. The BBS generated by the L-BBS-root is called L-BBS.

The distribution of the bandwidth is done by means of a two-step process that gives precedence to unsatisfied leaf classes. According to the first rule, a leaf class is allowed to transmit immediately if it is underlimit² and its L-BBS-root is underlimit as well. This prevents the L-BBS from consuming bandwidth reserved to other subtrees.

A second rule distributes excess bandwidth to the all the *unbounded* classes (according to the L-BBS they belong) when no classes are allowed to send according

¹ A bounded class is a class whose traffic that cannot exceed its allocated rate.

² A class is underlimit if its throughput (averaged over a certain period) does not exceed its allocated rate. See [1] for more details.

to the first rule. A class is allowed to get more bandwidth when it has a non-overlimit³ ancestor A⁴ at level i and there are no unsatisfied classes in its L-BBS at levels lower than i . This guarantees that the excess bandwidth is distributed inside the L-BBS; therefore an overlimit class is allowed to transmit provided that there is still bandwidth available in its L-BBS.

First rule allows a leaf class that is underlimit and bounded (that is the suggested configuration for real-time classes) to transmit without constraints, while an underlimit and unbounded leaf class can be delayed when its L-BBS-root is overlimit. Bounded leaf classes are never influenced by the behavior of other classes, while unbounded classes are. This does not represent a problem because an unsatisfied class that is delayed will be served as soon as its L-BBS-root becomes underlimit. Starvation does not occur and underlimit leaf classes are always able to reach their target rate; however the time-interval used to monitor their throughput must be larger than the time interval used to monitor bounded leaf classes.

Unfortunately, these rules do not guarantee that any BBS-root never becomes overlimit. An embedded L-BBS-root can be allowed to transmit (it is underlimit indeed), making the higher-level BBS-root overlimit. It follows that even D-CBQ does not respect perfectly the link-sharing structure; however this is the fee that has to be paid in order to have some classes that are always able to send (provided that they are respectful of their service rate). This fee is required to provide excellent support for guaranteed-bandwidth classes that, on the other hand, are not allowed to send more than their guaranteed rate.

2.2 Decoupling Bandwidth and Delay

This feature requires that the excess bandwidth will be distributed independently of the class priority. D-CBQ uses two distinct systems of cascading WRR schedulers; the first one is activated when bandwidth is distributed according to the first link-sharing rule; the second one distributes excess bandwidth and it is enabled when bandwidth is allocated according to the second rule. First WRR uses priorities (i.e. it gives precedence to high priority traffic) and it guarantees each class to be able to get its allocated rate; second WRR does not take care of priorities and it selects classes according to their share.

This mechanism is rather simple but effective: network managers can assign each user a specific value of bandwidth and delay and they will be certain that, whatever the priority is, excess bandwidth will be distributed evenly to all the currently active sessions.

Suspending Overlimit Classes

It has been widely recognized that link-sharing rules and delay guarantees cannot be met at the same time; therefore there are small time intervals in which one of them cannot be guaranteed. The decoupling between bandwidth and delay has to be

³ A class is non-overlimit when it does not exceed its rate, averaged over a certain period.

⁴ The class must be able to borrow from ancestor A. Basically, A can be either the root class (in case the L-BBS-root is unbounded) or a generic ancestor belonging to the L-BBS.

integrated with new rules that determine when a class has to be suspended (because either it or one of its parents is overlimit) and how long the suspension time will be.

First answer is based on the new link-sharing guidelines: a class is suspended when is not allowed to transmit according to the second rule. D-CBQ allows each class being suspended, whereas CBQ allows suspension of leaf classes only. In this case D-CBQ suspends the highest-level ancestor (i.e. the nearest to the root class) whom the class is allowed to borrow from and that is overlimit. Since ancestor class is suspended, all unbounded leaf classes that share this ancestor are no longer allowed to transmit. Bounded classes, of course, are still allowed to transmit (first link-sharing guideline).

Second question (*how long*) is based on the observation that a class (or a subtree, in the D-CBQ case) must be suspended for the time needed to be compliant to the allocated rate; hence the suspension time will be the one of the class that is being suspended, that is the ancestor class. It follows that the suspension time depends on the *extradelay* of the ancestor class; therefore it depends on the bandwidth allocated to the intermediate class instead of the one allocated to leaf classes. Generally speaking, suspension time depends on the "upper overlimit ancestor" instead of the leaf class.

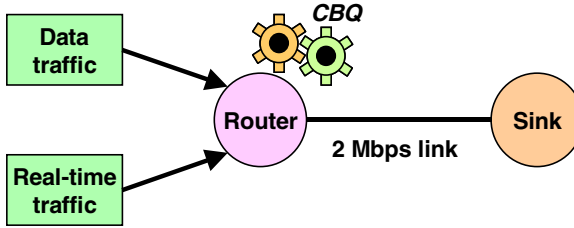


Fig. 2. Test topology.

3. Simulation Results

This Section presents some comparative results between CBQ (in the implementation that comes with *ns-2*) and D-CBQ through the *ns-2* simulator. Results have been obtained by defining two different test suites; the first one devoted to bandwidth and link-sharing properties and the second one devoted to the delay objectives⁵. Each test suite consists in several tests; each one has a different class configuration (link-sharing structure, borrow, priority) and it is structured in several simulations that have different incoming traffic and different bandwidth allocated to each class.

D-CBQ (as well as CBQ) is, by nature, a non-work conserving algorithm although it can be easily modified in order not to leave the output link idle when there are unbounded backlogged classes. For instance, results compare CBQ with two different

⁵ Simulations measure the scheduling delay, i.e. the time between the arrival of the packet and the time the scheduler finishes its transmission on the output link.

versions of D-CBQ, standard (described in Section 2) and *efficient*. Efficient D-CBQ (D-CBQe) looks like a work-conserving algorithm and it adds a new rule to D-CBQ. When no classes are allowed to transmit according to the link-sharing guidelines, D-CBQe sends a packet from the first unbounded and backlogged class it encounters. Therefore it is not a pure work-conserving algorithm because bounded classes are not able to exploit the output link idleness.

The efficient mechanism inserts a small degree of unfairness into D-CBQ. To keep the algorithm simple, the efficient process selects classes on a priority-based schedule; hence high priority classes can get more bandwidth. Moreover D-CBQ internal variables are not updated in case of a transmission due to this mechanism: the rational is that a class should not be punished for the bandwidth consumed when no classes are allowed to transmit.

Simulations use a simple topology (Fig. 2) composed by a single scheduler and several sources attached to it. Number of sources, their rate and their traffic pattern varies among the simulations. Sessions under test use CBR and Poisson sources because of their simplicity and their easiness to be controlled. Tests repeated using real sources (UDP/TCP sessions simulating real traffic) do not show any difference compared to previous ones.

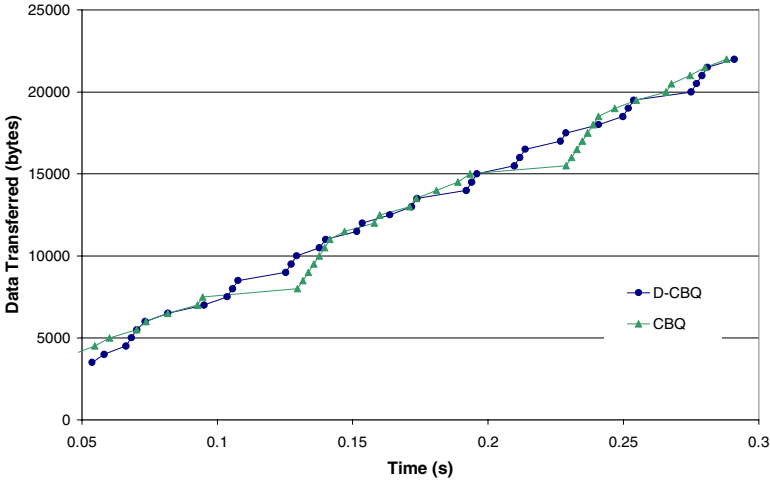


Fig. 3. New Link-Sharing guidelines.

New Link-Sharing Guidelines

This test shows that D-CBQ is able to guarantee each session a more predictable service. Fig. 3 shows a typical trace in which both CBQ and D-CBQ are able to provide the correct share over large-scale intervals. However this is not true over small scale: triangle trace shows that CBQ often suspends the class (large amount of time between two packets); this does not happen with D-CBQ.

Decoupling Bandwidth and Delay

D-CBQ has strong decoupling characteristics. For example Table 1 reports the results obtained with a simple 1-level hierarchy (all classes are children of the root class). Priority does not influence bandwidth in D-CBQ. In case of all classes competing for the bandwidth (first two tests), they are able to obtain the assigned share. Moreover, last test shows that CBQ assigns all the available bandwidth (not used by class B) to the high priority class; D-CBQ allocates the excess bandwidth to all classes proportionally to their share. The result is that D-CBQ looks more like a Weighted Fair Queuing than a Priority Queuing schema from this point of view and it is able to force malicious users not to exceed their allocated rate. Setting higher priorities (than means lower delays) in D-CBQ is no longer a way to obtain more bandwidth.

Table 1. Decoupling bandwidth and delay; all classes are allowed to borrow.

	Share	Priority	Traffic (Kbps)			
			In	CBQ out	D-CBQ out	Theor.
Class A	1%	LOW	100	55.25	21.46	20
Class B	99%	LOW	2000	1944.77	1978.56	1980
Class A	1%	HIGH	200	181.82	21.41	20
Class B	99%	LOW	2000	1818.19	1978.61	1980
Class A	10%	HIGH	2000	604.08	250.37	250
Class B	20%	LOW	---	---	---	---
Class C	70%	LOW	2000	1395.94	1749.65	1750

Quality Indexes

Previous tests are not able to validate for certain the goodness of D-CBQ from other points of view as well (delay, precision). Moreover, these tests have to be carried out using several different configurations; therefore data need to be summarized in order to display the results.

Comparison among different results is done using the following set of quality indexes:

$$Q^2 = \frac{1}{N} \cdot \sum_{i=1}^N \left(\frac{D_i - D_i^*}{D_i^*} \right)^2, \quad |Q| = \frac{1}{N} \cdot \sum_{i=1}^N \left| \frac{D_i - D_i^*}{D_i^*} \right|$$

where D_i^* is the expected test result (the theoretical value for link-sharing tests; the best obtained delay in the current simulation for delay tests), D_i is the actual result of the simulation and N is the number of simulations. The quadratic quality index (Q^2) highlights tests in which the behavior is significantly different from the expected value; therefore it is used to identify any idiosyncrasies between theoretical and real behavior. Vice versa, linear index ($|Q|$) is the relative difference of the simulation results compared to the theoretical ones and it can be used to show the precision of CBQ and D-CBQ against the expected result. Best results are obtained when these indexes tend to zero.

Link-sharing Test Suite Details

Main objective is to verify the ability to provide each class with its target bandwidth; therefore sources (CBR) exceed the rate allocated to that class. The link-sharing test suite is made up of 10 different tests that use three different class configurations (details in Appendix I); each test aims at the evaluation of specific aspects of the algorithm.

A brief summary of the results (details can be found in [5]) is shown in Table 2: quality index for CBQ is by far the worst of all. Results confirm that D-CBQ performs far better than the original algorithm, particularly in tests with borrowing enabled and different priorities among classes. |Q| shows that the difference between experimental results and theoretical one is greatly reduced from the 14.1% of CBQ to the 1.7% of D-CBQ.

An interesting point is that the efficient version of D-CBQ performs worse than the standard version. Efficient mode, in fact, inserts another trade-off between link utilization and precision parameters. For instance, a class could not be allowed to transmit at its target time when efficient mode is turned on because the scheduler could be busy servicing another packet.

Table 2. Link-sharing test results (values * 100).

Test	Quadratic Quality Index			Linear Quality Index		
	CBQ	D-CBQ	D-CBQe	CBQ	D-CBQ	D-CBQe
Test 1	0.3112	0.0111	0.0111	5.0110	0.8345	0.8345
Test 2	0.0000	0.0000	0.0000	0.0008	0.0008	0.0008
Test 3	0.2841	0.0646	0.0646	5.0140	2.0899	2.0899
Test 4	51.7934	0.0948	0.0948	30.5246	1.7590	1.7590
Test 5	0.2994	0.0721	0.0721	4.8157	2.2212	2.2212
Test 6	1091.2810	0.0886	0.0886	137.4755	1.6486	1.6486
Test 7	0.0008	0.0001	0.0001	0.2325	0.0671	0.0671
Test 8	0.0099	0.0001	0.0001	0.8036	0.0780	0.0780
Test 9	0.1672	0.0482	0.2211	3.5506	1.7568	2.7292
Test 10	0.3756	0.1706	0.3479	5.0937	2.5652	3.8744
Global	73.1666	0.0815	0.1784	14.0525	1.7387	2.3698

A small problem still remains: even D-CBQ is not able to transmit all the traffic when input sources are transmitting at their maximum rate. This is due to internal approximations. A good practice consists in a slight over-provisioning of the bandwidth allocated to that class; an in-depth analysis of this phenomenon is left to future studies.

Delay Test Suite Details

Delay test suite is made up of five tests that differ in the characteristics of the real-time sources. Simulations use both CBR and Poisson traffic for real-time sources, and a set of several VBR sessions for data traffic. Real-time sources transmit slightly less than the bandwidth allocated to their class; vice versa data traffic exceeds its allocation in order to make the output link congested. Three tests use a single CBR source for each session (each test has a different packet size); the fourth uses three

CBR sources with different packet sizes (120, 240, 480 bytes) for each session, while the last uses Poisson sources. Last two tests use a token-bucket limiter to regulate real-time sources and to control input pattern (and source's peak rate) with excellent accuracy. Twelve different simulations with different link-sharing structure, priority and borrowing characteristics compose each test.

This paper summarizes the results related to the maximum delay experienced by packets and the delay experienced by the 99% of them (99-percentile). Results are given only for the real-time traffic because best effort one exceeds its allocation; therefore delay has no significance. Detailed analyses for delay bounds are left for future work [6]; here only a brief summary is given.

Table 3. Maximum delay tests: results (values * 100).

Test	Quadratic Quality Index			Linear Quality Index		
	CBQ	D-CBQ	D-CBQe	CBQ	D-CBQ	D-CBQe
CBR1	4.940	0.001	0.008	1.566	0.009	0.033
CBR2	158.514	0.000	0.010	8.236	0.000	0.034
CBR3	175.056	0.000	0.004	8.765	0.000	0.025
CBR-M	0.268	0.097	0.107	0.176	0.111	0.099
PS	0.000	0.138	0.127	0.003	0.201	0.168
Global	67.756	0.047	0.051	3.749	0.064	0.072

Starting from the maximum experimented delay, results (shown in Table 3) confirm that D-CBQ outperforms CBQ in all tests. Packets flowing through CBQ have a maximum delay that is far larger than the one experimented by D-CBQ. CBQ performs better only in a few simulations and this is due to its different (and wrong) implementation of the WRR mechanism⁶.

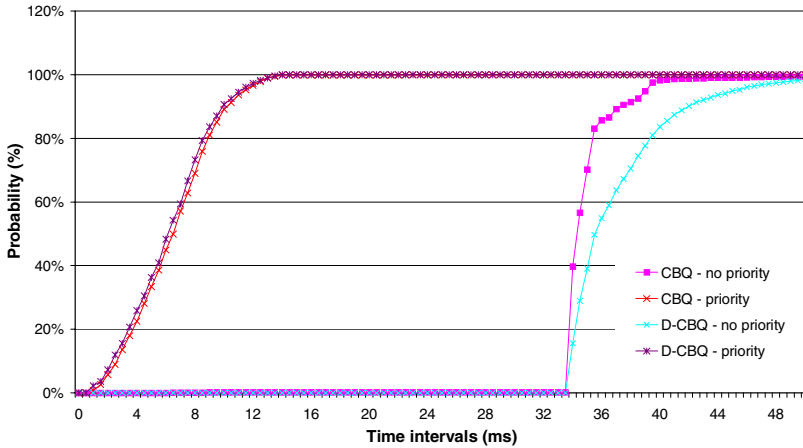
D-CBQ improvement concerning the 99-percentile delay bound (Table 4) is not so evident such as in the maximum delay bound. D-CBQ, however, still guarantees smaller delays: delays of the 99-percentile of the CBQ packets are 4 times larger than the D-CBQ ones. Results in which CBQ outperforms D-CBQ (like PS) are due (again) to the different WRR implementation.

Fig. 4 shows a typical distribution of the delay in CBQ and D-CBQ and it points out that delay distribution for high priority classes is almost the same, even if CBQ has a small percentage of packets that have significantly larger delays than D-CBQ.

⁶ A detailed analysis of the CBQ code shows that it does not implement correctly the WRR mechanism and it often sets a class allocation to zero arbitrarily instead of leaving it negative. This operation has a non-negligible impact on classes with large packets compared to their allocation: these classes do no longer need to wait several rounds before being able to transmit. Some configurations take particular advantage of that, hence CBQ may show better delay bounds.

Table 4. 99-percentile delay tests: results (values * 100).

Test	Quadratic Quality Index			Linear Quality Index		
	CBQ	D-CBQ	D-CBQe	CBQ	D-CBQ	D-CBQe
CBR1	4.940	0.001	0.008	1.566	0.009	0.033
CBR2	158.514	0.000	0.010	8.236	0.000	0.034
CBR3	175.056	0.000	0.004	8.765	0.000	0.025
CBR-M	0.268	0.097	0.107	0.176	0.111	0.099
PS	0.000	0.138	0.127	0.003	0.201	0.168
Global	67.756	0.047	0.051	3.749	0.064	0.072

**Fig. 4.** A typical cumulative distribution of the delay in CBQ and D-CBQ.

D-CBQe performances are never better than D-CBQ. Even if the efficient part of the algorithm affects unbounded classes only, this feature influences real-time classes (that are usually bounded) as well because it forces sometimes real-time classes to wait longer before transmitting a packet.

4. ALTQ Implementation

ALTQ implementation is almost the same as the ns-2 one. The most important difference is the inability to wake up a class exactly at its target time, since this would require setting a new timer event each time a class is suspended. This might overload the processing power of the machine; therefore the suspension time is approximated within discrete intervals based on the kernel timer (set to 1KHz on our machines) available in the BSD kernel. Other differences are related to real-world issues, for example the presence of output interface buffers (virtually a FIFO queue after the CBQ scheduler) and the possible mismatch between the theoretical wire speed and the real one (for instance, header compression, link-layer headers and more may alter the

real speed). First problem, pointed out in [3], is of great importance and is largely reduced in ALTQ 3.0 (several interface drivers have been modified). The second point does not have solutions at this moment and it results on some mismatch between some theoretical values (for example the packet departure time) and real ones.

ALTQ implementation has been validated using a subset of the test already used in ns-2 and results confirm the goodness of D-CBQ as well as in the simulations. The most important result, however, is that D-CBQ complexity is slightly more than CBQ one, although in presence of un-optimized code. This is evident primarily in the borrow tests in which D-CBQ has to check the status of the class' ancestors.

Table 5. ALTQ tests snapshot: maximum throughput (packets per second) on a Pentium 133 machine.

Test	Packet size (bytes)	CBQ	D-CBQ	Difference
No Borrow	40	7080	6599	-6.8%
	58	6992	6463	-7.6%
Borrow	40	6847	6186	-9.7%
	58	6691	6140	-8.2%

5. Conclusions

D-CBQ has been proved being a substantial improvement over CBQ. Its characteristics allow the deployment of this scheduler on networks with advanced requirements (hierarchical link-sharing, bandwidth guarantees, delay bounds).

Efficient D-CBQ has been shown being not worthy from the link-sharing and delay point of view. However further analyses are needed in order to evaluate the advantages of this algorithm on best effort traffic: we should expect some improvements in term of link utilization and throughput for these classes.

Next step will be a better characterization of D-CBQ from the viewpoint of the delay in order to give a mathematical indication of the maximum delay bound experimented by a D-CBQ session.

Source code for ns-2 and ALTQ, together with test script, is online at the Author's website.

Acknowledgements

The author thanks Salvatore Iacono, Giordana Lisa and Kenjiro Cho for many discussions about CBQ internals. Best thanks also to Ivan Ponzanelli and Lucio Mina for their insightful help in testing and validating the prototypes, Panos Gevros, Mario Baldi and Jon Crowcroft for their comments.

This work has been partially sponsored by Telecom Italia Lab, S.p.A., Torino (Italy).

References

- [1] Sally Floyd and Van Jacobson, Link Sharing and Resource Management Models for Packet Networks, IEEE/ACM Transaction on Networking, Vol. 3 No. 4, August 1995.
- [2] Ion Stoica, Hui Zhang, T. S. Eugene Ng, A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Service, in Proceedings of SIGCOMM '97 September 1997.
- [3] Fulvio Rizzo and Panos Gevros, Operational and Performance Issues of a CBQ router, ACM Computer Communication Review, Vol. 29 No 5, October 1999.
- [4] The VINT Project, UCB/LBNL/VINT Network Simulator - ns (version 2). Available at <http://www-mash.cs.berkeley.edu/ns/>.
- [5] Ivan Ponzanelli, Garanzie di servizio con schedulers di tipo Class Based Queuing, Laurea Thesis, Politecnico di Torino, July 2000. In Italian.
- [6] Fulvio Rizzo, Delay Guarantees in D-CBQ, Draft Paper, Politecnico di Torino, April 2001.
- [7] Fulvio Rizzo, Decoupling Bandwidth and Delay Properties in Class Based Queuing, in Proceedings of the Sixth IEEE Symposium on Computers and Communications (ISCC 2001), July 2001

Appendix I

This appendix presents the details of the link-sharing and delay test suites.

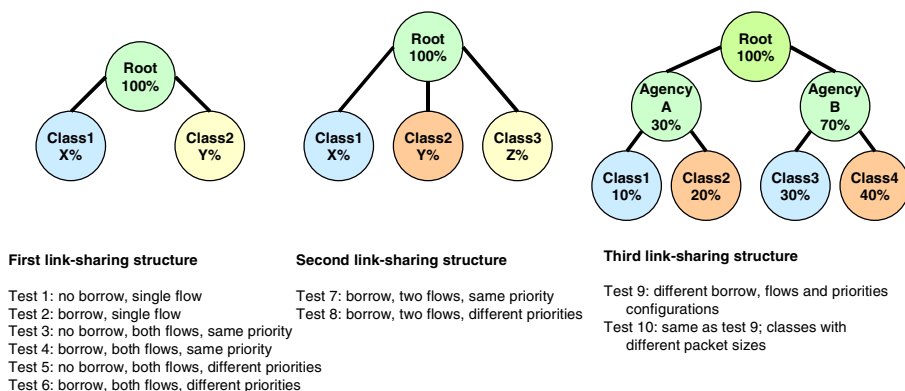


Fig. 5. Link-sharing test suite: link-sharing structure.

The link-sharing test suite is made up of 10 different tests that use the three different class configurations shown in Fig. 5. Each test aims at the evaluation of specific aspects of the algorithm. In detail:

- Test 1: precision of the traffic carried by a single class, taken in isolation
- Test 2: ability to exploit all the link bandwidth by a single class, taken in isolation
- Tests 3, 4: ability to share correctly the bandwidth among peer classes; tests have been performed with and without borrowing
- Tests 5, 6: same as tests 3 and 4; classes have different priorities
- Tests 7, 8: ability to share the excess bandwidth correctly; classes can have either equal or different priorities
- Tests 9,10: ability to respect the imposed bandwidth among classes with different priorities, borrow configuration, incoming traffic; Test 10 repeats the same simulations using different packet sizes among classes. Configuration details are shown in Table 6, as well as the expected throughput of each class.

Delay test-suite is similar to the previous one: it consists in five link-sharing hierarchies, with different class configuration and traffic. A summary of the test characteristics is reported in Fig. 6.

Table 6. Details of tests 9 and 10 (throughput in Kbps).

Simulations		Classes					
		A	1	2	B	3	4
1	Priority Borrow/Traffic Expected throughput	LOW Y/---	LOW Y/Y 200	LOW Y/Y 400	LOW Y/---	LOW Y/N ---	LOW Y/Y 1400
2	Priority Borrow/Traffic Expected throughput	LOW N/---	LOW N/Y 200	LOW N/Y 400	LOW Y/---	LOW Y/N ---	LOW Y/Y 1400
3	Priority Borrow/Traffic Expected throughput	LOW N/---	LOW N/Y 200	LOW N/Y 400	LOW Y/---	LO Y/N	LOW Y/N
4	Priority Borrow/Traffic Expected throughput	LOW Y/---	LOW Y/Y 200	LOW Y/Y 400	LOW Y/---	LOW Y/Y ⁷ 200	LOW Y/Y 1200
5	Priority Borrow/Traffic Expected throughput	LOW N/---	LOW Y/Y 200	LOW Y/Y 400	LOW Y/---	LOW Y/N	LOW Y/Y 1400
6	Priority Borrow/Traffic Expected throughput	HIGH Y/---	HIGH Y/Y 200	HIGH Y/Y 400	LOW Y/---	LOW Y/N	LOW Y/Y 1400
7	Priority Borrow/Traffic Expected throughput	HIGH Y/---	HIGH Y/Y 200	LOW Y/Y 400	HIGH Y/---	LOW Y/N	HIGH Y/Y 1400
8	Priority Borrow/Traffic Expected throughput	HIGH N/---	HIGH Y/Y 200	HIGH Y/Y 400	LOW Y/---	LOW Y/N	LOW Y/Y 1400
9	Priority Borrow/Traffic Expected throughput	HIGH N/---	HIGH Y/Y 200	LOW Y/Y 400	HIGH Y/---	LOW Y/N	HIGH Y/Y 1400

⁷ This test uses an on-off source, with 33% activity period.

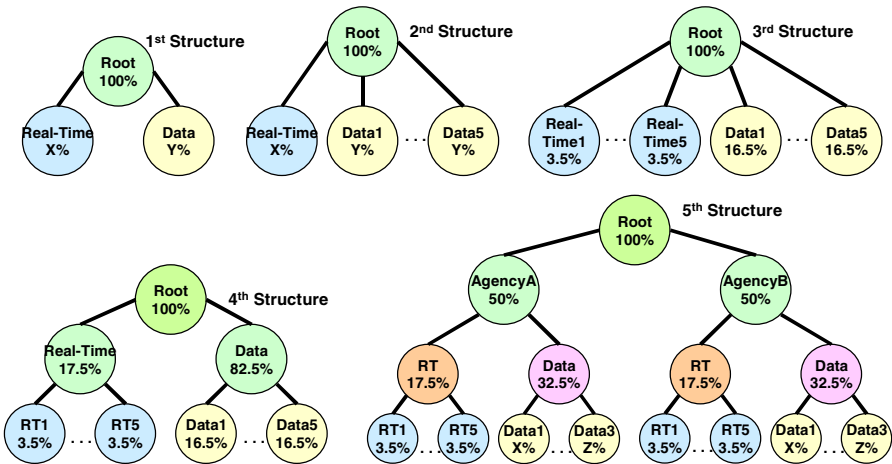


Fig. 6. Delay tests: the link-sharing structure.

A Performance Study of Explicit Congestion Notification (ECN) with Heterogeneous TCP Flows

Robert Kinicki and Zici Zheng

Worcester Polytechnic Institute
Computer Science Department
Worcester, MA 01619, USA

Abstract. This paper compares the simulated performance of RED routers and ECN routers. The results show that ECN provides better goodput and fairness than RED for heterogeneous TCP flows. When demand is held constant, increasing the number of flows generating the demand has a negative effect on performance. ns-2 simulations with many flows demonstrate that the bottleneck router's marking probability must be aggressively increased to provide good ECN performance. These experiments suggest that an adaptive version of ECN should provide better performance than ECN.

1 Introduction

With increased World Wide Web traffic has come heightened concern about Internet congestion collapse. Since the first congestion collapse episode in 1986, several variants of TCP (Tahoe, Vegas, Reno and NewReno) have been developed and evaluated to provide host-centric mechanisms to combat high packet loss rates during heavy congestion periods. Additionally, researchers have proposed new congestion avoidance techniques for Internet routers. While the initial concept was to use packet loss at FIFO routers to signal congestion to the source, the resulting drop-tail behavior failed to provide adequate early congestion notification and produced bursts of packet drops that contribute to unfair service.

Since the introduction of Random Early Detection (RED) [6] in 1993, researchers have proposed a variety of enhancements and changes to router management to improve congestion control while providing fair, best-effort service. Although RED has outperformed drop-tail routers in several simulation and test-bed experiments [1], [4], [5], [8], [9], [12], Christainsen et al [3] have demonstrated that tuning RED for high performance is problematic when one considers the variability of Internet traffic.

RED has been shown to be unfair when faced with heterogeneous flows [10], [14] and the recommended RED parameter settings [6] are not aggressive enough in heavy congestion generated by a large number of flows [3], [5], [8].

Concern over reduced performance on the Internet during traffic bursts such as Web flash crowds helped spawn the IETF recommendation [2] for new active

queue management techniques that provide early congestion notification to TCP sources. Several research studies [1], [7], [8], [9] have reported better performance for Explicit Congestion Notification (ECN) when compared against RED. These results add support to the Internet draft "Addition of ECN to IP" [16]. However, most of these studies cover only a limited portion of the traffic domain space. Specifically, little attention has been given to evaluating the effects of a large number of concurrent flows. Although a couple of these studies consider fairness among competing homogeneous flows, ECN behavior with heterogeneous flows has not been thoroughly studied.

This paper presents results from a series of ns-2 simulations comparing the ability of RED and ECN to provide fair treatment to heterogeneous flows. The goal of this report is to add to the existing information on ECN behavior specifically with regard to the impact of the number of flows, the effect of ECN tuning parameters on performance, and the effectiveness of ECN's congestion warnings when many flows cause the congestion. The results of this study provide insight into a new active queue management scheme, AECN, Adaptive ECN.

Section 2 briefly defines a few measurement terms and reviews previous ECN studies to provide context for our experiments. Section 3 discusses experimental methods. The next section analyzes the simulated results and the final section includes concluding remarks.

2 Definitions and Background

The performance metrics used in this investigation include delay, goodput and two ways to evaluate fairness. The delay is the time in transit from source to destination and includes queuing time at the router. Goodput differs from throughput in that it does not include retransmitted packets in the count of packets successfully arriving at the receiver. Given a set of flow throughputs

$$(x_1, x_2, \dots, x_n)$$

Jain's fairness index [13] is defined in terms of the following function

$$f(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n (x_i)^2}$$

A second form of fairness introduced in section 4 focuses on the difference between the maximum and minimum average goodput for groups of heterogeneous flows [8].

Random Early Detection (RED) [6] utilizes two thresholds (*min_th*, *max_th*) and an exponentially-weighted average queue size, *ave_q*, to add a probabilistic drop region to FIFO routers. *max_p* is a RED tuning parameter used to control the RED drop probability when *ave_q* is in the drop region. The drop probability increases linearly towards *max_p* as *ave_q* moves from *min_th* to *max_th*. When *ave_q* reaches *max_th*, RED switches to a deterministic (100%) drop probability. *max_th* is set below the actual queue length to guarantee drops that signal router congestion before the physical queue overflows.

Explicit Congestion Notification (ECN) [12],[15] marks a packet (instead of dropping) when ave_q is in the probabilistic drop region. In the deterministic drop region, ECN drops packets just as RED does. We briefly consider an ECN variant, ECNM, that marks packets in the deterministic region.

Lin and Morris [10] define fragile TCP flows as those emanating from sources with either large round-trip delays or small send window sizes and robust flows as having either short round-trip delays or large send windows. This delineation emphasizes a flow's ability to react to indications of both increased and decreased congestion at the bottleneck router. Our experiments simulate three distinct flow groups (fragile, average, and robust flows). These flows differ only in their end-to-end round-trip times (RTTs). To simplify the analysis, the maximum sender window is held fixed at 30 packets throughout this investigation.

Floyd's original ECN paper [7] shows the advantages of ECN over RED using both LAN and WAN scenarios with a small number of flows. Bagal et al [1] compare the behavior of RED, ECN and a TCP rate-based control mechanism using traffic scenarios that include 10 heterogeneous flows. They conclude that RED and ECN provide unfair treatment when faced with either variances due to the RTTs of the heterogeneous flows or variances in actual flow drop probabilities. Focusing on a window advertising scheme (GWA), Gerla et al [8] compare GWA, RED, and ECN in scenarios with up to 100 concurrent flows. Using the gap between maximum and minimum goodput as a fairness measure, they show that ECN yields better fairness than RED for homogeneous flows. Salim and Ahmed [17] use Jain's fairness to compare ECN and RED performance for a small number of flows. Their results emphasize that max_p can significantly effect performance. The ns-2 experiments discussed in this paper combine and extend these results.

3 Experimental Methods and Simulation Topology

This study uses the newest version of Network Simulator from UCB/LBNL, **ns-2** [11], to compare the performance of ECN and RED routers with TCP Reno sources. The simulation network topology (shown in Figure 1) consists of one router, one sink and a number of sources. Each source has a FTP connection

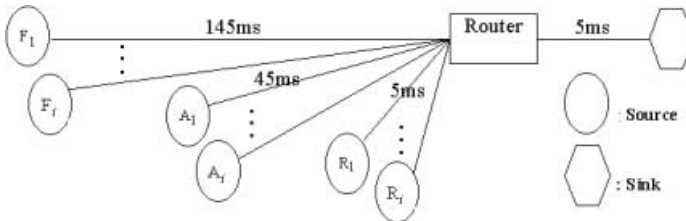


Fig. 1: Simulation Topology

feeding 1000-byte packets into a single congested link. The bandwidth of the bottleneck link is 10Mbps with a 5 ms delay time to the sink. The one-way link delays for the fragile, average and robust sources are 145 ms, 45 ms and 5 ms respectively. Thus, the fragile, average and robust flows have round-trip times of 300 ms, 100 ms and 20 ms when there is no queuing delay at the router.

All simulations ran for 100 simulated seconds. Half the flows were started at time 0 and the other half were started at 2 seconds. The graphs presented exclude the first 20 seconds to reduce transient startup effects. The router for all simulations has a *min_th* of 5 packets and a physical queue length of 50 packets. Except for the maximum send window size of 30 packets, all other parameters use the ns-2 default values.

4 Results and Analysis

A series of ns-2 experiments were run such that the cumulative traffic flow into the heavily congestion router **remains fixed** at 600 Mbps even though the number of flows is varied across simulations. In all cases, the number of flows is equally divided among the three flow categories. Thus, 15 flows in the graphs implies 5 fragile, 5 average and 5 robust flows each with a 40 Mbps data rate whereas a graph point for 120 flows implies a simulation with 40 fragile, 40 average and 40 robust flows each with a 5 Mbps data rate. Simulations were run with the total number of flows set at 15, 30, 60, 120, 240, 480 and 600 flows.

Figure 2 gives ECN and RED goodput with the number of flows varying from 15 to 600. ECN with *max_p* = 0.5 provides the best goodput in all cases except 15 flows. In the other router configurations there is a large drop in goodput beginning at 64 flows. Figure 3 presents the delay for ECN and RED with *max_p* = 0.5. This figure shows the clear advantage robust flows have with respect to delay, but more importantly it demonstrates that the ECN goodput improvement from Figure 2 is offset by a small increase in the one-way delay for ECN.

Figures 4 and 5 track the effect of varying *max_p* and *max_th* in simulations with 30 and 120 flows respectively. Figure 4 shows that *max_th* has little effect on goodput above *max_p* = 0.2. In Figure 5 where 120 flows provide the same flow demand as 30 flows in Figure 3, ECN with *max_p* = 0.5 and *max_th* = 30 yields the highest goodput and there is no *max_p* setting for RED that works well.

Figure 6 employs Jain's fairness to quantify RED and ECN behavior. ECN is fairer than RED in almost all situations. Since perfect fairness has a Jain's fairness index of 1, it is clear that as the number of flows goes above 120 none of the choices prevent unfairness. The fact that ECN with *max_p* = 0.1 is fairest at 30 flows while *max_p* = 0.5 is the fairest at 60 and 120 flows implies the marking probability could be dynamically adjusted for ECN based on a flow count estimator. The unfairness at a high number of flows can be partially attributed to a lockout phenomenon where some flows are unable to get any packets through the congested router for the duration of the simulation. Locked out flows begin to appear for both RED and ECN above 120 flows.

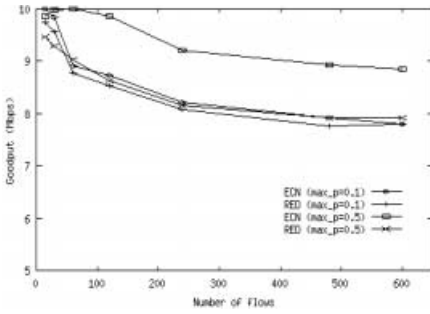


Fig. 2: RED and ECN Goodput, $max.th=30$

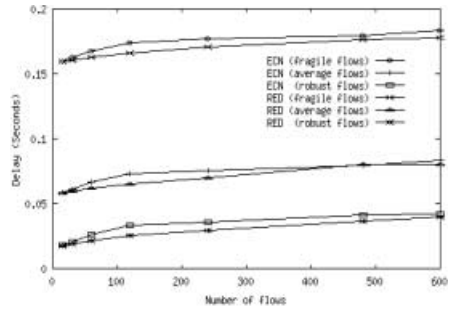


Fig. 3: RED and ECN Delay, $max.p=0.5$, $max.th=30$

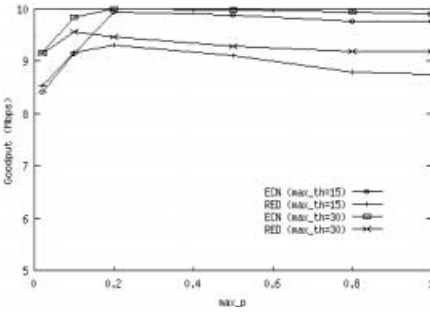


Fig. 4: Goodput with 30 flows

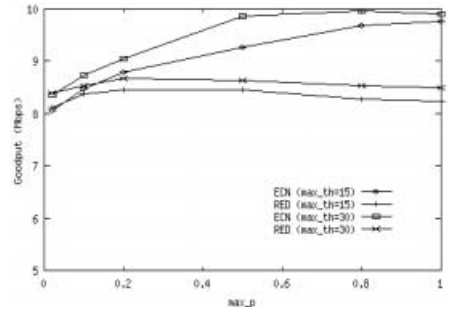


Fig. 5: Goodput with 120 flows

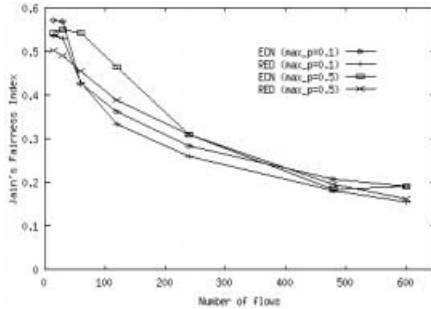
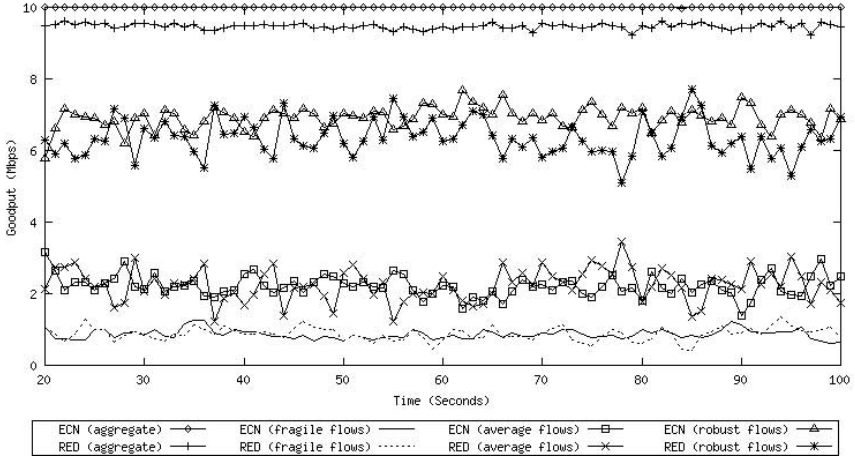
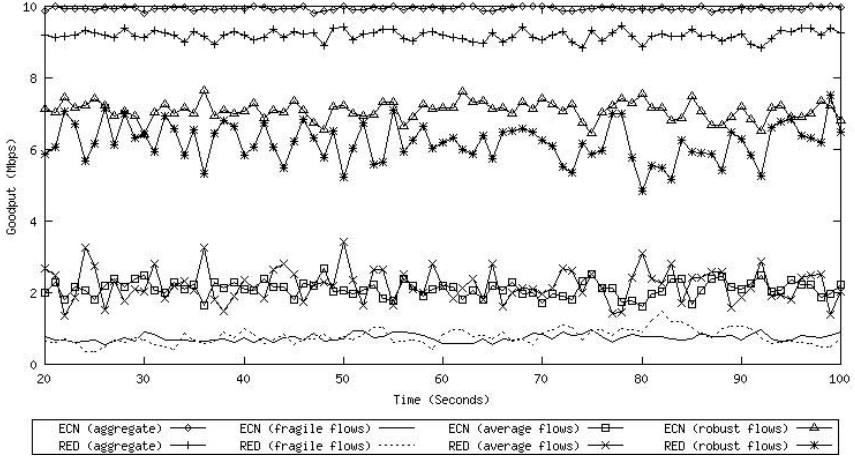


Fig. 6: RED and ECN Fairness, $max.th=30$

Figures 7 through 9 provide a visual sense of max-min fairness for RED and ECN via the gap between the average goodputs for the three flow groups.

Aggregate goodput in these graphs is the sum of the fragile, average, and robust goodputs. ECN provides better aggregate goodput than RED in all three graphs, but the difference is most pronounced in Figure 9 where the traffic is generated by 120 flows. Figure 7 and 8 differ only in an increase of $max.p$ from 0.2

Fig. 7: Goodput Distribution, 30 flows, $max\_p=0.2$, $max\_th=30$ Fig. 8: Goodput Distribution, 30 flows, $max\_p=0.8$, $max\_th=30$

to 0.8. The more aggressive ECN marking in Figure 8 provides better goodput for robust flows than RED. However this change does not reduce the goodput gap between robust and fragile flows. Figure 9 keeps $max\_p = 0.8$ but simulates 120 flows. Although aggregate goodput remains relatively unchanged for ECN in Figure 9, the goodput for the robust flows goes down while the goodput of the average and fragile flows increase slightly. This implies that an adaptive ECN that uses different values of max_p for heterogeneous flows can provide improvement in the visual max-min fairness. RED goodput is adversely affected by more flows.

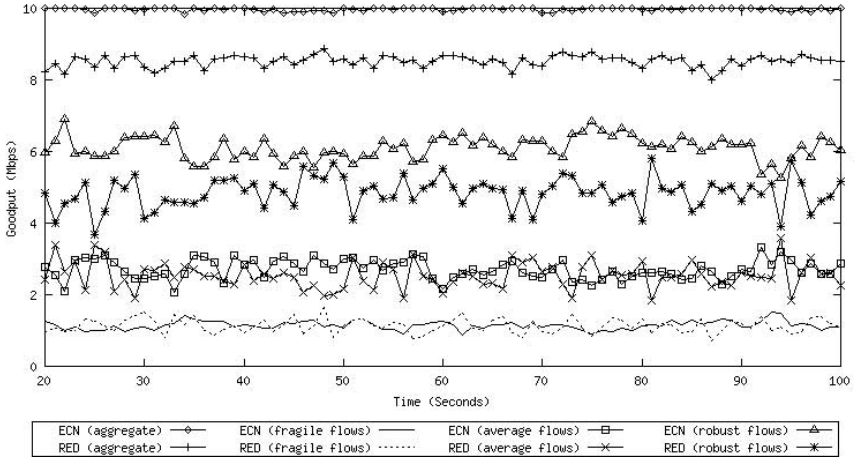


Fig. 9: Goodput Distribution, 120 flows, $max\_p=0.8$, $max\_th=30$

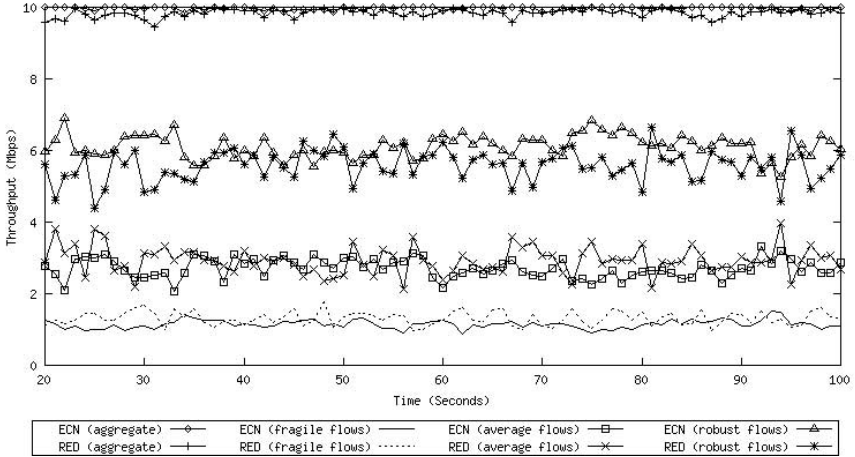


Fig. 10: Throughput Distribution, 120 flows, $max\_p=0.8$, $max\_th=30$

The significance of using goodput instead of throughput as a performance metric can be clearly seen in Figures 9 and 10. Because goodput excludes re-transmissions, RED has 15% lower goodput than ECN in Figure 9. Since RED drops and ECN marks, the RED drops trigger more TCP retransmissions. This effect is completely hidden in Figure 10 where aggregate RED throughput is only slightly lower than aggregate ECN throughput.

Figure 11 compares ECN with ECNM. Recall ECNM differs from standard ECN in that ECNM marks packets when the average queue size exceeds max_th and drops packets only when the router queue overflows. The figure shows that

ECN provides better goodput except at small values of max_p and that ECNM appears quite sensitive to the max_th setting.

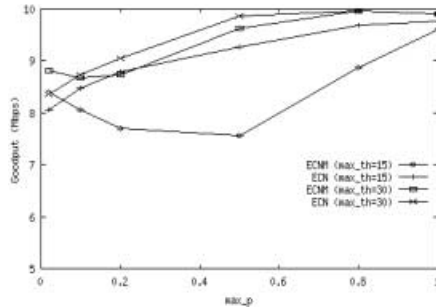


Fig. 11: ECN and ECNM Goodput with 120 flows

5 Conclusions and Future Work

This paper reports on a series of ns-2 simulations that compare ECN and RED performance with heterogeneous TCP flows. Generally ECN provides better goodput and is fairer than RED. The results show that for fixed demand the performance of both mechanisms decreases as the number of flows increases. However, ECN with an aggressive max_p setting provides significantly higher goodput when there are a large number of heterogeneous flows. ECN also had a higher Jain's fairness index in the range of flows just below where flow lockouts occurred.

In the simulations studied neither RED nor ECN strategy were fair to fragile and average flows. These results suggest that if congestion control is to handle Web traffic consisting of thousands of concurrent flows with some degree of fairness then further enhancements to ECN are needed. We are currently conducting simulations with an adaptive version of ECN that adjusts max_p based on the round-trip time of a flow and an estimate of the current number of flows in each flow group.

References

1. P. Bagal, S. Kalyanaraman, B. Packer, Comparative Study of RED, ECN and TCP Rate Control, Technical Report (Mar 1999).
2. B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309 (Apr 1998).

3. M. Christainsen, K. Jeffay, D. Ott, F.D. Smith, Tuning RED for Web Traffic, Proc. SIGCOMM 2000, Stockholm (Sep 2000) 139-150.
4. W. Feng, D. Kandlur, D. Saha and K. Shin, A Self-Configuring RED Gateway, Proc. INFOCOM 1999, New York (Mar 1999) 1320-1328.
5. W. Feng, D. Kandlur, D. Saha and K. Shin, Blue: A New Class of Active Queue Management Algorithms, U. Michigan CSE-TR-387-99 (Apr 1999).
6. S. Floyd and V. Jacobson, Random Early Detection Gateways for Congestion Control, IEEE/ACM Transactions on Networking (Aug 1993) 397-413.
7. S. Floyd, TCP and Explicit Congestion Notification, ACM Computer Communication Review, vol. 24, no. 5 (Oct 1994) 10-23.
8. M. Gerla, R.L. Cigno, S. Mascolo, W. Weng, Generalized Window Advertising for TCP Congestion Control, UCLA Tech Report 990012, 1999.
9. S. Kunniyur and R. Srikant, End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks, Proc. INFOCOM 2000, Tel-Aviv (Mar 2000) 1323-1332.
10. D. Lin and R. Morris, Dynamics of Random Early Detection, ACM Computer Communication Review, vol. 27, no. 4 (Oct 1997) 127-138.
11. <http://www.isi.edu/nsnam/ns/doc/index.html>.
12. T.J. Ott, T.V. Lakshman, and L.H. Wong, SRED: Stabilized RED, Proc. INFOCOM 1999, New York (Mar 1999) 1346-1355.
13. L.L. Peterson and B.S. Davie, Computer Networks, A Systems Approach, 2nd Ed., Morgan Kaufmann, San Francisco, 2000.
14. A.M. Raghavendra and R.E. Kinicki, A Simulation Performance Study of TCP Vegas and Random Early Detection, Proc. 18th IEEE International Performance, Computing, and Communications Conference, Phoenix (Feb 1999) 169-176.
15. K.K. Ramakrishnan, and S. Floyd, A Proposal to Add Explicit Congestion Notification (ECN) to IP, RFC 2481 (Jan 1999).
16. K.K. Ramakrishnan, S. Floyd, D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, Internet Draft (Nov 2000).
17. J. H. Salim and U. Ahmed, Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks, RFC 2884 (Jul 2000).

Diffusion Model of RED Control Mechanism

Rachid Laalaoua¹, Stanisław Jędrus², Tülin Atmaca¹, and Tadeusz Czachórski²

¹ Institut National des Télécommunications
9, rue Charles Fourier, 91011 Evry Cedex, France
{Laalaoua,Atmaca}@etna.int-evry.fr
<http://www.int-evry.fr>

² IITiS PAN
ul. Bałtycka 5, 44-100 Gliwice, Poland
{Stanley,Tadek}@iitis.gliwice.pl

Abstract. We present a diffusion model of a network node controlled by RED mechanism used to indicate congestion but not to delete packets. Diffusion approximation allows us to study the dynamics of flow changes introduced by this mechanism in a more efficient way than simulation. After introducing some basic notions on diffusion approximation and on our approach to solve diffusion equations analytically or numerically, we present a closed loop model of flow control and investigate the influence of delay and of control parameters on performance of the system. Also FECN/BECN scheme is considered: flow remains constant within an interval of fixed length and is changed in next interval if the number of marked packets during the interval is above a certain threshold. Diffusion results are validated by simulation.

1 Introduction

Diffusion approximation has been proven to be an efficient tool to analyse transient states in various traffic control mechanisms, e.g. leaky bucket or threshold queue [6] and sliding window [10]. Here, we adapt it to model dynamics of Random Early Detection (RED) algorithm which received much attention, e.g. [8,11] and is recommended by Internet Engineering Task Force as a queue management scheme for rapid deployment [3], as it turns out that end-to-end TCP window-based congestion control mechanism is not sufficient to ensure Internet stability and should be supplemented by router-based schemes. Its performance was modelled with the use of simulation or Markov chains [7]. The principle of RED mechanism is to start discarding packets with a specified probability *before* the buffer becomes full, opposite to the principles of Tail Drop mechanism. The probability of discarding packets is given by a specified drop function, see Fig. 1. The argument η of this function is a weighted moving average of queue length: $\eta := (1 - w)\eta + wn$ where w is a constant and n is current queue length upon arrival of a new customer. Explicit Congestion Notification is an extension proposed to RED which marks a packet instead of dropping it. Since ECN marks packets before congestion actually occurs, this is useful for protocols like TCP that are sensitive to even a single packet loss. Upon receipt of a congestion

marked packet, the TCP receiver informs the sender (in the subsequent ACK) about incipient congestion which will in turn trigger the congestion avoidance algorithm at the sender. In this paper, we study the performance of ECN mechanism using diffusion approximation and simulation. We compare it also with a virtual circuit model in Frame Relay or ATM networks with closed-loop feedback control mechanism. The model is based on the use of the FECN/BECN scheme.

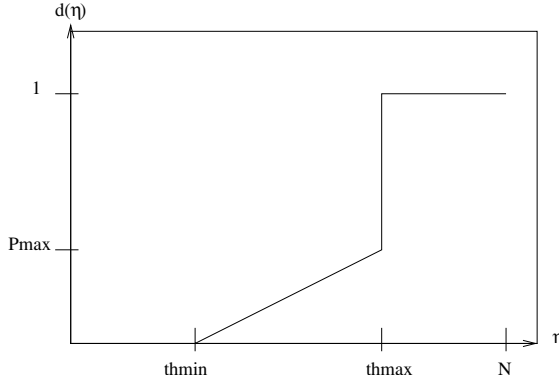


Fig. 1. RED drop function

After introducing some basic notions on diffusion approximation and on our approach to solve diffusion equations analytically (Section 2) or numerically (Section 3), we present a closed loop model of flow control based on RED mechanism (Section 4) and its extension to FECN/BECN scheme (Section 5). Section 6 gives several numerical results presenting validation of diffusion models and a study of the influence of some control parameters on the network performance. Section 7 presents conclusions.

2 Transient Solution to G/G/1/N Queue: Diffusion Approximation, Analytical Approach

Diffusion approximation [12] represents the number $N(t)$ of customers in a queue at a time t by the value of a diffusion process $X(t)$; the density function $f(x, t; x_0)$ of this process is given by diffusion equation

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} \quad (1)$$

and approximates the queue distribution: $f(n, t; n_0) \approx p(n, t; n_0) = \Pr[N(t) = n | N(0) = n_0]$. Diffusion parameters α , β represent the characteristics of input flow and service time distribution. For a G/G/1 or G/G/1/N model where $1/\lambda$,

σ_A^2 are mean and variance of interarrival time distribution $A(x)$ and $1/\mu$, σ_B^2 are mean and variance of service time distribution $B(x)$, they are chosen as, e.g. [9]

$$\beta = \lambda - \mu, \quad \alpha = \sigma_A^2 \lambda^3 + \sigma_B^2 \mu^3 = C_A^2 \lambda + C_B^2 \mu. \quad (2)$$

Diffusion model with two barriers and instantaneous returns of G/G/1/N station was proposed by Gelenbe in [9] where the steady-state solution of the model was given. The diffusion process has two barriers at $x = 0$ and $x = N$. When the process comes to $x = 0$, it stays there for a time representing the idle period and then jumps immediately to $x = 1$ starting new busy period. Similarly, when the process attains $x = N$, it stays at the barrier for a time representing the period when the queue is full and then jumps to $x = N - 1$. To obtain transient solution, we proceed similarly as in [5] representing the density function of diffusion process with instantaneous returns by a superposition of densities of another diffusion process: the one with absorbing barriers at $x = 0$ and $x = N$.

Consider a diffusion process with two absorbing barriers at $x = 0$ and $x = N$, started at $t = 0$ from $x = x_0$. Its probability density function $\phi(x, t; x_0)$ has well known form, see e.g. [4,6]. If the initial condition is defined by a function $\psi(x)$, $x \in (0, N)$, $\lim_{x \rightarrow 0} \psi(x) = \lim_{x \rightarrow N} \psi(x) = 0$, then the pdf of the process has the form $\phi(x, t; \psi) = \int_0^N \phi(x, t; \xi) \psi(\xi) d\xi$. The probability density function $f(x, t; \psi)$ of the diffusion process with elementary returns is composed of the function $\phi(x, t; \psi)$ which represents the influence of the initial conditions and of a spectrum of functions $\phi(x, t - \tau; 1)$, $\phi(x, t - \tau; N - 1)$ which are pd functions of diffusion processes with absorbing barriers at $x = 0$ and $x = N$, started at time $\tau < t$ at points $x = 1$ and $x = N - 1$ with densities $g_1(\tau)$ and $g_{N-1}(\tau)$:

$$f(x, t; \psi) = \phi(x, t; \psi) + \int_0^t g_1(\tau) \phi(x, t - \tau; 1) d\tau + \int_0^t g_{N-1}(\tau) \phi(x, t - \tau; N - 1) d\tau. \quad (3)$$

Densities $\gamma_0(t)$, $\gamma_N(t)$ of probability that at time t the process enters to $x = 0$ or $x = N$ are

$$\begin{aligned} \gamma_0(t) &= p_0(0) \delta(t) + [1 - p_0(0) - p_N(0)] \gamma_{\psi,0}(t) + \int_0^t g_1(\tau) \gamma_{1,0}(t - \tau) d\tau \\ &\quad + \int_0^t g_{N-1}(\tau) \gamma_{N-1,0}(t - \tau) d\tau, \\ \gamma_N(t) &= p_N(0) \delta(t) + [1 - p_0(0) - p_N(0)] \gamma_{\psi,N}(t) + \int_0^t g_1(\tau) \gamma_{1,N}(t - \tau) d\tau \\ &\quad + \int_0^t g_{N-1}(\tau) \gamma_{N-1,N}(t - \tau) d\tau, \end{aligned} \quad (4)$$

where $\gamma_{1,0}(t)$, $\gamma_{1,N}(t)$, $\gamma_{N-1,0}(t)$, $\gamma_{N-1,N}(t)$ are densities of the first passage time between points specified in the index, e.g.

$$\gamma_{1,0}(t) = \lim_{x \rightarrow 0} \left[\frac{\alpha}{2} \frac{\partial \phi(x, t; 1)}{\partial x} - \beta \phi(x, t; 1) \right]. \quad (5)$$

The functions $\gamma_{\psi,0}(t)$, $\gamma_{\psi,N}(t)$ denote densities of probabilities that the initial process, started at $t = 0$ at the point ξ with density $\psi(\xi)$ will end at time t by entering respectively $x = 0$ or $x = N$.

Finally, we may express $g_1(t)$ and $g_N(t)$ with the use of functions $\gamma_0(t)$ and $\gamma_N(t)$:

$$g_1(\tau) = \int_0^\tau \gamma_0(t) l_0(\tau - t) dt, \quad g_{N-1}(\tau) = \int_0^\tau \gamma_N(t) l_N(\tau - t) dt, \quad (6)$$

where $l_0(x)$, $l_N(x)$ are the densities of sojourn times in $x = 0$ and $x = N$; the distributions of these times are not restricted to exponential ones. Laplace transforms of Eqs. (4,6) give us $\bar{g}_1(s)$ and $\bar{g}_{N-1}(s)$; the Laplace transform of the density function $f(x, t; \psi)$ is obtained as

$$\bar{f}(x, s; \psi) = \bar{\phi}(x, s; \psi) + \bar{g}_1(s) \bar{\phi}(x, s; 1) + \bar{g}_{N-1}(s) \bar{\phi}(x, s; N - 1). \quad (7)$$

Probabilities that at the moment t the process has the value $x = 0$ or $x = N$ are

$$\bar{p}_0(s) = \frac{1}{s} [\bar{\gamma}_0(s) - \bar{g}_1(s)], \quad \bar{p}_N(s) = \frac{1}{s} [\bar{\gamma}_N(s) - \bar{g}_{N-1}(s)]. \quad (8)$$

The Laplace transforms $\bar{f}(x, s; \psi)$, $\bar{p}_0(s)$, $\bar{p}_N(s)$ are inverted numerically following Stehfest algorithm [13].

3 Diffusion Approximation — Numerical Approach

The approach presented above gives transient solution to a diffusion equation with constant parameters α and β . If the parameters are changing with time, we should define the time periods (e.g. of the length of one mean service time) where they may be considered constant and solve diffusion equation within these intervals separately; transient solution obtained at the end of an interval serves as the initial condition for the next one. If the input stream or service times depend on the queue length, the diffusion parameters depend also on the value of the process: $\alpha = \alpha(x, t)$, $\beta = \beta(x, t)$. In this case also the diffusion interval $x \in [0, N]$ is divided into subintervals of unitary length and the parameters are kept constant within these subintervals. For each time- and space-subinterval with constant parameters, transient diffusion solution is obtained. The equations for space-intervals are solved together with balance equations for probability flows between neighbouring intervals. For more complex models it is convenient to solve a diffusion model entirely numerically. A method of lines [14] has been adapted to fit the case of diffusion equation. The basis of this method, which is sometimes called the generalized method of Kantorovich, is substitution of finite differences for the derivatives with respect to one independent variable, and retention of the derivatives with respect to the remaining variables. This approach changes a given partial differential equation into a system of partial differential equations with one fewer independent variable.

They were the following implementation problems:

1. instantaneous returns introduce Dirac delta functions, which in turn have to be approximated and cause singularities in the integrated function,
2. a sum of the integral of probability density function and probabilities that process is in boundary barriers should be constant and equal to 1,
3. parameters of equation are changing; since they are dependent on a temporary state of a model, their values (especially α) can be small enough to cause a lack of stability of computations.

The first problem has been solved by approximation of Dirac delta function with a rectangular impulse function dx wide and $1/dx$ high where dx denotes integration step on x axis. The second obstacle was avoided using a conservative, centered, 2-order scheme:

$$p_{k+1,n} - p_{k,n} = \left((p_{k,n+1}\beta_{k,n+1} - p_{k,n-1}\beta_{k,n-1}) \frac{\Delta\tau}{2\Delta x} - 0.5(p_{k,n+1}\alpha_{k,n+1} + p_{k,n-1}\alpha_{k,n-1} - 2p_{k,n}\alpha_{k,n}) \frac{\Delta\tau}{(\Delta x)^2} \right). \quad (9)$$

The probability mass which moves at a step k into the barriers is computed as $p_{k+1,0}\Delta x$ and $p_{k+1,N}\Delta x$ where N denotes the location of right barrier. This mass is added to probabilities p_0 and p_N that process is in barrier either 0 or N respectively and $p_{k+1,0}$ and $p_{k+1,N}$ are assigned with 0 (barriers are absorbing the mass). The solution of the third mentioned problem is to approximate the movement of the probability mass with fluid flow approximation whenever $\alpha(x, t)$ is too small. Unfortunately, to conserve the probability mass it would require to recompute whole integration step in t axis every time such an action appears (we should also apply a special integration scheme in the nearest neighbourhood of such a point). Thus having in mind that the case that α is close to zero is exceptional and does not occur too frequently the equation is simply renormalized.

The stability of the integration scheme was evaluated using the Von Neumann test and the presented method is not stable if $\Delta\tau > (\Delta x)^2/4$. However, the estimation of stability region is complex, since parameters of equation are dependent.

4 Diffusion Model of RED

Consider a G/G/1/N queue with time-dependent input. In diffusion model we cannot distinguish the moments of arrivals. We simply consider the queue size in intervals corresponding to mean interarrival times, $\Delta t = 1/\lambda$. The traffic intensity λ remains fixed within an interval. When λ changes, also the length $1/\lambda$ of the interval is changed. We know the queue distribution $f(x, t; x_0)$ at the beginning of an interval. This distribution also gives us the distribution $r(x, t)$ of response time of this queue. Let us suppose that T is the total delay between the output of RED queue and the arrival of flow with new intensity, following changes introduced by the control mechanism based on RED.

At the beginning of an interval i we compute the mean value of the queue: $E[N_i] = 1 + \sum_{n=1}^N p_i(n)n$, the weighted mean

$$\eta_i = (1 - w)\eta_{i-1}p_i(0) + [1 - p_i(0)] [(1 - w)\eta_{i-1} + wE[N_i]]$$

and then we determine on this basis, using the drop function $d(\eta)$ of RED as in Fig. 1, the probability of tagging the packet to announce the congestion. Following it, we determine the changes of traffic intensity:

$$\lambda_{new} = [1 - d(\eta)] (\lambda_{old} + \Delta\lambda) + d(\eta)\lambda_{old}/a$$

but only in the case when the last change was done earlier than a predefined *silent period*. The increase of flow is additive with $\Delta\lambda$ and the decrease of flow is multiplicative with constant $1/a$. This λ_{new} will reach the queue after the round trip time $E[N_i] \cdot 1/\mu + T$.

5 Diffusion Model of FECN/BECN Scheme

Consider a slightly different control scheme: the traffic intensity λ remains fixed within a control interval D . For each $\Delta t = 1/\lambda$ (supposed moments of new arrivals), we compute $E[N_i]$ and, if $E[N_i] \geq \text{threshold}$, the counter is increased. At the end of the interval the value of λ_{new} is obtained, according to the ratio of marked packets, that means to the ratio of the counter content to the value $D \cdot \lambda$ (the supposed number of packets that arrived during interval D)

$$\lambda_{new} = [1 - p] (\lambda_{old} + \Delta\lambda) + p\lambda_{old}/a$$

where:

$$p = \begin{cases} 0 & \text{if } \text{marked packet ratio} \leq \text{predefined value, e.g. 0.5} \\ 1 & \text{otherwise} \end{cases}$$

This mechanism corresponds to forward (or backward) explicit congestion notification (FECN/BECN) scheme.

6 Numerical Results

Fig. 2 presents the studied model and Figs. 3 – 8 display some typical results. We choose constant service time $1/\mu = 1$ as a time unit (t.u.). The buffer capacity is $N = 40$. The values of other model parameters, i.e. delay T , thresholds thr_{min} , thr_{max} , as well as control interval D and threshold thr for FECN/BECN scheme are given in captions of figures.

Fig. 3 displays, in logarithmic and linear scales, examples of RED queue distributions for two different times: when the queue is relatively lightly loaded ($t = 30$ t.u.) and when it is overcrowded ($t = 60$ t.u.). The curves in logarithmic scale show that, in spite of 100 000 repetitions of the experiment, simulation

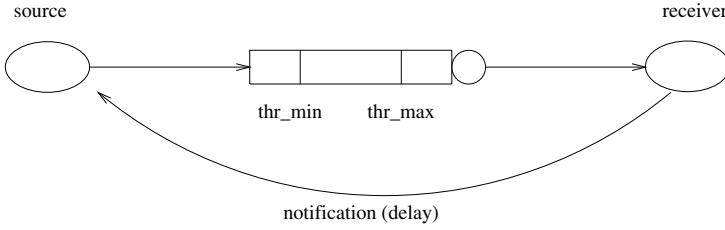


Fig. 2. Investigated system with RED mechanism (in case of FECN/BECN mechanism, the queue has only one threshold)

model has still problems with determination of small values of probabilities. The same remark may be made for Fig. 6 presenting loss as a function of time – the simulation cannot give small loss probabilities while diffusion is able to furnish very small values (which are naturally approximative).

Figs. 4, 5 give diffusion and simulation results for mean queue length as a function of time and for the resulting from RED mechanism time-dependent traffic throughput. Poisson input stream (Fig. 4) and constant (deterministic) stream (Fig. 5) are considered.

Fig. 7 presents long-range performance of RED mechanism. In the left figure the overall loss ratio, taken in the time interval of $T = 10\,000$ t.u. length, is presented as a function of delay for two sets of parameters: (1) $w = 0.002$, $p_{max} = 0.02$ and (2) $w = 0.2$, $p_{max} = 0.5$; silent period = 0. The constant w is the weight with which the current value of queue is taken into consideration at η . For the first set of parameters the loss is high (the mean queue value is nearly 35, see right figure) and practically does not depend on delay. For the second set of parameters, the loss is lower and its growth with the value of delay is distinctly visible. Simulation results are obtained in two ways: they represent either real loss, i.e. the ratio of lost packets to the whole number of arrived packets, or the overall probability of full queue, $\sum_{t=1}^T p(N, t)/T$ where $t = 1, 2, 3, \dots$ denotes consecutive slots of unitary length. Although the input stream is Poisson, the both results are not the same: as there is permanent transient state, the property of PASTA does not hold. Diffusion approximation represents naturally the second approach. For the first set of parameters, diffusion curve is between both simulation results; for the second set of parameters probabilities of full queue obtained by diffusion and simulation are very close. In the right figure the mean queue length and the corresponding moving average η , which is the argument of RED function $d(\eta)$, are displayed for the same two sets of parameters (1), (2) as in left figure. Only simulation results are given. It is visible that the oscillations observed in previous figures which are due to initial conditions, especially the ones for moving average η , attenuate with time.

Fig. 8 relates to the evaluation of FECN/BECN mechanism. In left figure the throughput given by diffusion model is compared with simulation results; in right figure we see the evolution of simulated FECN mean queue for two different thresholds compared to the RED mean queue (one of those displayed in Fig. 7).

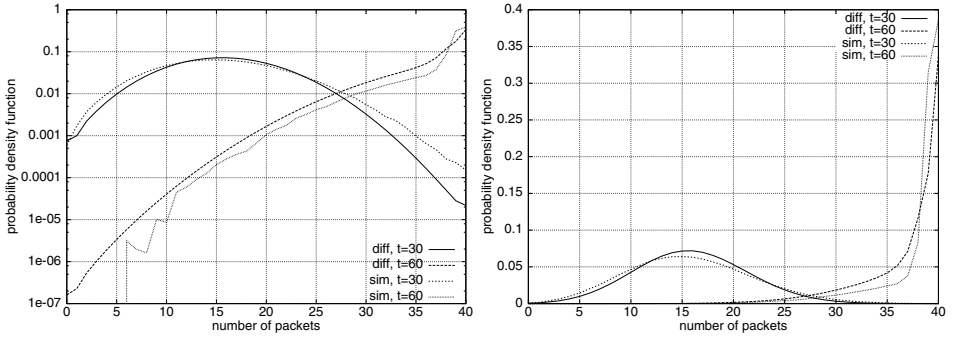


Fig. 3. Queue distribution at $t = 30$ t.u. and $t = 60$ t.u., model parameters: $th_{min} = 25$, $th_{max} = 35$, buffer capacity $N = 40$, initial condition $N(0) = 0$, Poisson input, *silent period* = 0, *delay* = 5 t.u.; logarithmic and linear scale, diffusion and simulation results

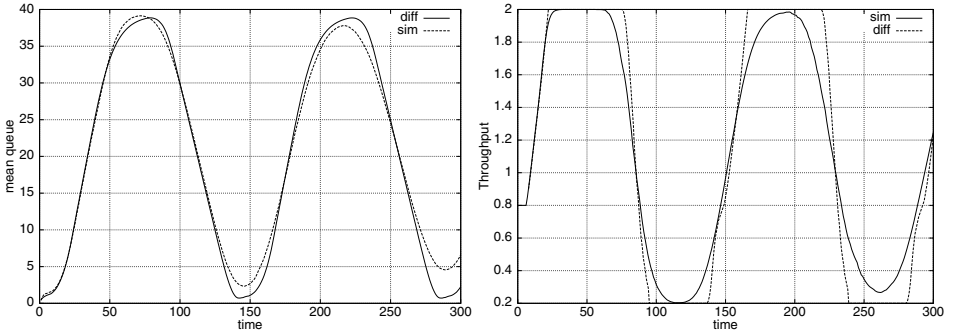


Fig. 4. Mean queue length and changes of flow, diffusion and simulation results, Poisson input, parameters as in Fig. 3

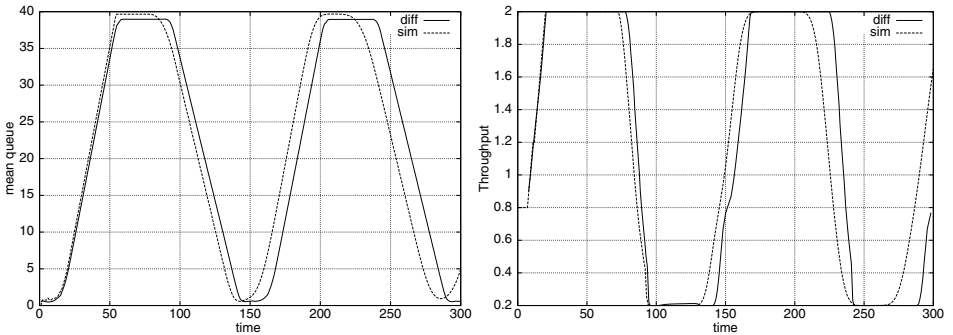


Fig. 5. Mean queue and changes of flow in simulation and diffusion model, deterministic input flow, other parameters as in Fig. 3

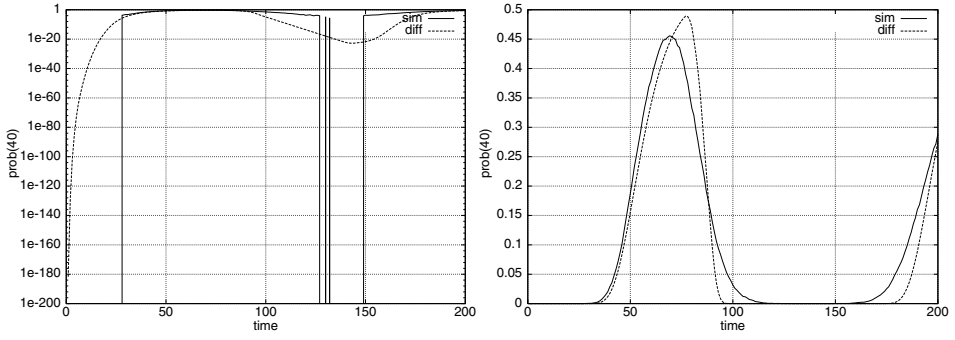


Fig. 6. Loss as a function of time, parameters as in Fig. 3, diffusion and simulation results; logarithmic and linear scale

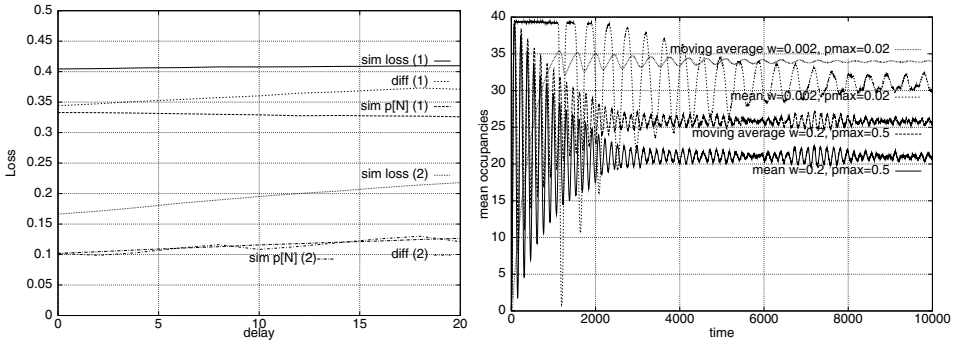


Fig. 7. Long scale performance of RED with Poisson input; *left*: overall loss ratio as a function of delay for two sets of parameters: (1) $w = 0.002$, $p_{max} = 0.02$ and (2) $w = 0.2$, $p_{max} = 0.5$, *silent period* = 0, simulation and diffusion results; *right*: mean queues and moving averages as a function of time, the same as in *left* sets of parameters (1) and (2), *silent period* = 5 t.u., *delay* = 5 t.u.; simulation results

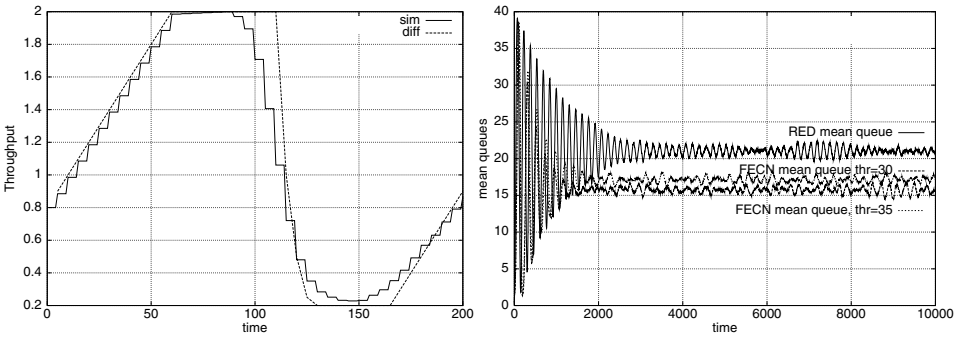


Fig. 8. FECN/BECN performance; *left*: throughput as a function of time, simulation and diffusion results; *right*: comparison of mean queues FECN ($D = 5$, $thr = 30$ or $thr = 35$, *delay* = 0) and RED ($w = 0.2$, $p_{max} = 0.5$, *silent period* = *delay* = 5 t.u.)

7 Conclusions

In this paper, we study with the use of diffusion approximation the impact of RED control mechanism on the network performance. Diffusion approximation has several natural advantages: flexibility and easiness to analyse transient states, to unify separate models into queueing networks, to consider different queue disciplines and control mechanisms, and to include in models time-varying input streams. Here, it gives us a tool to investigate the dynamics of flow changes introduced by RED and FECN/BECN mechanisms and to see the influence of parameters at RED drop function, of notification delay, etc. It is less costly as simulation, especially when small probabilities are to be determined. As demonstrate numerical examples, it gives reasonable results for considered cases. However, one should be aware of made approximations, all related numerical problems and the need of careful software implementation.

References

1. Ahmed, U., Salim, J. H.: Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks. In manuscript, Carlton University
2. Atmaca, T., Czachórski, T., Pekergin, F.: A Diffusion Model of the Dynamic Effects of Closed-Loop Feedback Control Mechanisms in ATM Networks. 3rd IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, Ilkley, UK, 4-7th July 1995
3. B. Braden et al.: Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC 2309, April 1998
4. Cox, R.P., Miller, H., D.: The Theory of Stochastic Processes. Chapman and Hall, London 1965
5. Czachórski, T.: A method to solve diffusion equation with instantaneous return processes acting as boundary conditions. Bulletin of Polish Academy of Sciences, Technical Sciences **41** (1993), no. 4
6. Czachórski, T., Pekergin, F.: Diffusion Models of Leaky Bucket and Partial Buffer Sharing Policy: A Transient Analysis, in: D. Kouvatsos (Editor), ATM Networks, Performance Modelling and Analysis, Chapman and Hall, London 1997
7. Laalaoua, R., Czachórski, T., Atmaca, T.: Markovian Model of RED Mechanism. CCGrid'2001, Brisbane Australia, 15-18 May 2001
8. Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, vol. 1, pp. 397-413, August 1993
9. Gelenbe, E.: On Approximate Computer Systems Models. J. ACM, vol. 22, no. 2, 1975
10. Jouaber, B., Atmaca, T., Pastuszka, M., Czachórski, T.: Modelling the Sliding window Mechanism. The IEEE International Conference on Communications, ICC'98, pp. 1749-1753, Atlanta, Georgia, USA, 7-11 czerwiec 1998
11. May, M., Bonald, T., Bolot, J.-Ch.: Analytic Evaluation of RED Performance. Proc. of Infocom'2000, Tel Aviv, Israel
12. Newell, G. F.: Applications of Queueing Theory. Chapman and Hall, London 1971
13. Stehfest, H.: Algorithm 368: Numeric inversion of Laplace transform. Comm. of ACM, vol. 13, no. 1, pp. 47-49, 1970
14. Zwingler, D.: Handbook of Differential Equations. Academic Press, Boston 1989, pp. 623-627

A Simple Admission Control Algorithm for IP Networks

Keith Kim, Petros Mouchtaris, Sunil Samtani,
Rajesh Talpade*, and Larry Wong

Telcordia Technologies, 445, South Street, Morristown, NJ 07960, USA
`rrt@research.telcordia.com`*
Phone: 973 829 4261 Fax: 973 829 5888

Abstract. We present a simple and scalable admission control algorithm for improving the Quality of Service (QoS) in Internet Service Provider (ISP) networks. The algorithm is novel in that it does not make any assumptions regarding the underlying transport technology (works at the IP layer), requires simple data structures and is low in operational complexity, can handle IP network topology changes efficiently, and can help identify congested links in the network. We have verified the working of this algorithm by simulation for arbitrary IP network topologies, and have found it to be successful in performing admission control and identifying congested links after route changes.

1 Introduction

IP networks are increasingly being used to transport different types of traffic such as voice, video, web and transactions. These different traffic types have different QoS requirements from the network, such as IP packet delay, jitter, loss and loss distribution. The Internet Engineering Task Force is developing the Differentiated Services approach ([1]) to provide the required QoS to the different traffic types, or classes. Briefly, the scalable DiffServ approach involves restricting fine granularity conditioning and marking of traffic at network edges, and processing traffic aggregates in the network core. The network is assumed to be provisioned to support the QoS requirements of each traffic class by assuring the necessary resources (bandwidth and buffer space) in the network.

While the Differentiated Services approach takes care of segregating different traffic classes and assuring minimum resources to each, it is also necessary to ensure that load within each traffic class remains within bounds such that the QoS requirements of that class are satisfied. An admission controller can be used to limit the load placed on each class. In the absence of such a mechanism, it is possible that the addition of a new flow within a class can result in degraded service to existing flows in that class. Additionally, the admission controller can be used to identify potentially over-loaded links, which can then be re-provisioned to support the higher traffic load. Local information is not sufficient

* Corresponding author. Authors listed alphabetically.

to perform this task efficiently, and complete network resource information is required, as illustrated in Figure 1. No capacity is assumed available between nodes D and E, while capacity is available between nodes C and D. So flows from A to F should be admitted while flows from A to G should be denied. Local information at A is not sufficient to make the correct admission control decision. It is important to note at this point that some traffic classes such as best-effort may not need admission control.

We present a simple and scalable admission control algorithm for improving the Quality of Service (QoS) in ISP networks for traffic classes that need an admission control strategy. The algorithm is novel in that it does not make any assumptions regarding the underlying transport technology such as ATM or MPLS, and works at IP layer. It considers global network resource information in the admission control decision process, and is scalable since it requires simple data structures to maintain this information. The algorithm is low in operational complexity. Coupled with network feedback, it can handle IP network route changes efficiently by updating network resource information and can help identify congested links. We have verified the working of this algorithm by simulation for arbitrary IP network topologies.

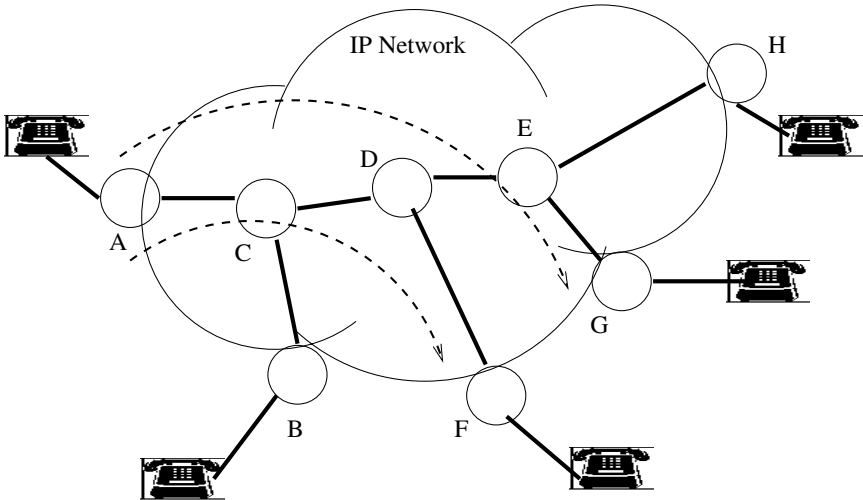


Fig. 1. Local vs. Global Admission Control

2 The Admission Control Algorithm

2.1 Assumptions and Terminology

Figure 2 illustrates the ISP network assumed in this work. It is composed of multiple Point of Presences (PoPs), interconnected by a core network. As has been mentioned before, specific link layer technologies are not required in the

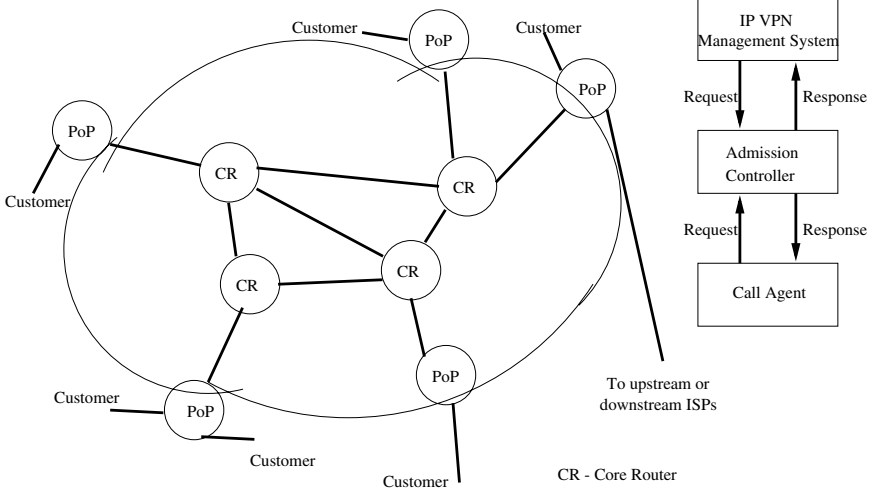


Fig. 2. Representative ISP IP Network Architecture

PoP or core. A path is unidirectional and is defined by a pair of PoPs. Thus the network comprises of multiple paths, and each path comprises of the sequence of links on the path. A link is defined as an IP level hop connecting a pair of routers. IP-level connectivity between PoPs is assumed to be provided by a dynamic IP routing protocol. Hence IP route changes can result in a change in the set of links comprising a path. The ISP is assumed to have defined the traffic classes that they plan to support, and provisioned their network accordingly. QoS is assumed to be a non-issue within each PoP, and the focus here is on assuring QoS *between* the ISP PoPs. A traffic flow is defined as a stream of IP packets between a pair of PoPs.

The admission control algorithm is intended for use within an ISP network (domain). It can potentially reside in an admission controller that processes requests from applications such as Voice over IP (VoIP) and IP Virtual Private Networks (VPN). For example, in case of VoIP, a Call Agent ([2]) application can interface with the admission controller for permission to setup VoIP calls between end-hosts. A traffic class for VoIP would be needed in this case. In case of IP VPN, the admission controller can reside in an IP VPN network management system that receives customer requests for IP VPN setup, and determines the feasibility of a request being supported by the ISP's existing network. The request could be in the form of a traffic matrix specifying the bandwidth requirements for each class between customer sites. Each site would be considered connected to an ISP Point of Presence. A positive decision in both the VoIP and IP VPN cases could result in the pertinent edge routers being configured to permit the requested flows. The algorithm is designed to be used only during the setup phase of a flow. In case of VoIP, this would be while the call is being setup, or resources are being reserved for a large number of calls

between a pair of PoPs. In case of IP VPN, this would be while a VPN setup request is being processed by the IP VPN management system.

2.2 Data Structures

The admission control algorithm uses two main data structures, *path* and *link*, to track network resource information. These data structures are designed to reduce the amount of information required and the processing complexity. Unless mentioned otherwise, information is maintained and processed only for those classes that require admission control. The path data structure includes information, for each path, about set of links comprising each path, the allocated bandwidth for each class, and current status of the path ("route changed" flag). The link data structure includes information, for each link, about the available bandwidth on the link (set initially to provisioned bandwidth) for each class, and a field indicating the "changed" bandwidth (explained later) on the link for each class.

2.3 Algorithm

The admission control algorithm is used in processing a request that results in the addition or deletion of flows to or from the network. It is thus designed to be used on the *control* path of a flow during the setup phase, and not on the *data* path. Hence IP routers do not need to perform the admission control or be aware of such a process. They can focus on data forwarding, in keeping with the spirit of the DiffServ approach. The admission control algorithm is described below.

Flow Admission Request

Input: Pair of PoPs for traffic flow, bandwidth requirement of each class in flow.

Output: Accept/Reject decision

Algorithm:

```
for each traffic class {
    if (required bandwidth >= available bandwidth for this class
        on any link on path)
        reject request for this class;
        /* can identify link for upgrade/re-provisioning and
           re-run algorithm */
    else {
        admit this class;
        update path data structure to reflect allocated bandwidth;
        update link data structure to modify available bandwidth
        for each link on path;
    }
}
```

Flow Release Request

Input: Pair of PoPs for traffic flow, traffic classes, allocated bandwidth for each class.

Algorithm:

```
for each traffic class
    update link data structure to reflect released bandwidth for
    each link on path;
```

2.4 Improving the Link Capacity Estimate

The above algorithm relies on the accuracy of its estimate of link capacity for correct admission control decisions. In the presence of route changes, the traffic over network links can vary. It is essential for the algorithm to track these changes and incorporate the change in available capacity of links in its link data structure. The above algorithm can be significantly improved by periodically providing feedback from the network about current IP connectivity on all the paths in the network (i.e. between each pair of PoPs). This feedback can be generated by tools such as *traceroute* between each pair of PoPs, and sending the IP route information to the admission controller. The admission controller can use this information to identify changed paths (set “route changed” flag), update the set of links comprising each path in the path data structure, recompute the available capacity on the links and identify links that may be congested as a result of IP route change. At the end of every feedback period, the following algorithm can be used to update the data structures as a result of IP route changes.

Link Capacity Estimation

```
for all paths in path data structure {
    if path has changed {
        for all new links on path
            add bandwidth allocated to path to “changed”
            bandwidth in link data structure for each class;
        for all deleted links on path
            subtract bandwidth allocated to path from “changed”
            bandwidth in link data structure for each class;
    }
}
```

The effect of route changes on available link bandwidth is now handled by subtracting “changed” bandwidth from available bandwidth in the link data structure for each class. The links with their “changed” bandwidth exceeding the available bandwidth can be considered to be congested. These links can be flagged by the admission controller for re-provisioning to handle the additional

traffic load. The “changed” bandwidth field provides an indication of the magnitude of change in load on the links during a feedback period. If this functionality is not required, the available bandwidth can be directly updated in the link capacity estimation algorithm. Otherwise, the following algorithm can be used to update the available capacity on the links.

```

/* continued from Link Capacity Estimation */

for all links in link data structure {
    if ("changed" bandwidth >= available bandwidth)
        flag link as congested;
    subtract “changed” bandwidth from available bandwidth
    and store in available bandwidth in link data structure;
    reset “changed” bandwidth to zero;
}

```

3 Discussion

The admission request and release algorithms are both $O(\text{average path length} * \text{number of traffic classes})$ in complexity. Since the average number of IP hops between pairs of PoPs (path length) is typically small ([3]), and the number of traffic classes that need admission control are also expected to be small to ensure proper network resource distribution between the classes, the processing overhead imposed by the algorithm is reasonable. The link capacity estimation component, which can be used after every feedback period, is $O(\text{number of paths} * \text{average path length} * \text{number of traffic classes})$ in complexity. Also, the algorithm requires connectivity data from the network for every pair of PoPs. While this component can introduce more overhead than the previous ones, the frequency of the feedback can be reduced if the load on the admission controller or other system components is found to be outside acceptable bounds. The feedback frequency should be greater than the frequency of route changes *within* ISP domains, and a feedback period on the order of 15 minutes should typically be adequate.

The admission control process can consider the latest set of links comprising a path, and their available bandwidth, due to the network feedback. The link data structure does not keep track of specific paths using each link, which simplifies the flow release process. IP route changes can be handled efficiently since path and link information is decoupled into separate data structures. We have verified the working of the algorithm using a simulation with arbitrary IP network topologies. The C-based simulation used multiple pre-generated topologies for a simple IP network, with each topology representing the connectivity (and consequent route changes) between all pairs of PoPs after failure of a specific set of links. Flow admission and deletion requests were generated and processed for each PoP pair at the end of each feedback period, representing all the requests

generated during that feedback period. The simulation can be run for specified number of feedback periods, with a new topology being chosen randomly at the end of each period. The algorithm was successful in performing admission control and identifying congested links after route changes. We have incorporated the algorithm in our Bandwidth Broker prototype, which is described in [4], [5] and [6]. The prototype Bandwidth Broker is being used in an experimental IP network [7] that is part of the InternetII Qbone, where it interfaces with various applications such as VoIP and medical image transfer.

References

1. Blake, S., et al, "An Architecture for Differentiated Services," IETF RFC 2475, December 1998.
2. Christian Huitema, et al., "An Architecture for Residential Internet Telephony Service," IEEE Network Magazine, May/June 1999.
3. Boardwatch Magazine's Digest of Internet Service Providers.
4. Internet2-Qbone Bandwidth Broker, <http://www.merit.edu/working.groups/i2-qbone-bb/>
5. Kim, K., et al, "A Bandwidth Broker Architecture for VoIP QoS," submitted to conference, 2000.
6. Kim, K., et al, "QoS Provisioning for VoIP in Bandwidth Broker Architecture: A Simulation Approach," in Proceedings of Communication Networks and Distributed Systems modeling and Simulation Conference (CNDs), 2001.
7. HUBS Project, <http://www.hubs.org/medical.html>.

An Architecture for a Scalable Broadband IP Services Switch

Manju V. Hegde, Mort Naraghi-Pour, Jean Bordes,
Curtis Davis, Otto Schmid, and Monier Maher

Celox Networks, Inc.,
940 West Port Plaza, St. Louis, MO 63146, USA
(TEL) 314-439-8300, (FAX) 1-314-317-8949
(mvhegde, mpour)@celoxnetworks.com

Abstract. Core routers cannot provide the generalized and flexible computing power for supporting the substantial amount of processing needed for services on the Internet, where configuration and processing is needed per user for a large number of users. In this paper we propose an architecture for a highly scalable broadband IP services switch which supports per user services for a large number of users, provides for isolation between users and between groups of users, provides security and authentication services through tunneling protocols such as *L2TP* and *IPSec* and supports traffic management through policing and shaping for fairness among different users. In this architecture the considerable time-consuming, repetitive processing is performed in specialized hardware while general purpose processing units are used for the computing power needed to enable the desired services. Load sharing and load balancing is used to distribute the computations between the general-purpose processing units.

1 Introduction

Historically, quality of service on the Internet has been what is called “best effort”. Best effort implies that the network only provides one class of service and that all the connections are treated equally when congestion arises. While this might have been acceptable for traditional Internet applications, it is highly inadequate for new real-time applications such as voice, video, webconferencing and other interactive applications. Yet these applications provide the greatest potential for the growth of services on the Internet.

To support such services, network nodes must support differentiated per-user and per-packet processing. However, while the high-bandwidth core routers that are currently under deployment are optimized for performing large numbers of fast routing lookups, they are not expected to provide generalized and flexible computing power for supporting the substantial amount of processing needed for, among other things, per user and per packet processing.

In order to emulate the services that subscribers currently avail of from different networks, and in order to enhance those offerings on a single ubiquitous

network, a network infrastructure needs to support the following general requirements.

1. **Quality of Service:** In order to support real-time applications, the Internet must provide differentiated quality of service (QoS). The Diffserv working group of IETF has defined the general architecture for differentiated services focusing on the definition and standardization of per-hop forwarding behaviors (PHBs) [1]. Network routers that provide differentiated services enhancements to IP must use the diffserv codepoint (DSCP) in the IP header to select a PHB for the forwarding treatment of that packet. In addition, packets that do not carry the diffserv codepoint but require differentiated service must first be marked with the appropriate DSCP before being forwarded. Marking of these packets is achieved using a classifier which must look up layer-3 and layer-4 headers in the packets. In order to support end-to-end QoS, it is almost mandatory to support multi-protocol label switching (MPLS) [5] and traffic engineering extensions to OSPF or IS-IS, which carry topological information necessary for traffic engineering.
2. **Multi-protocol support:** There are competing “last mile” technologies today which provide transport to the user for delivering packets to the “edge” of the Internet. To complete the communication, these packets need to be formatted to allow them to enter the Internet cloud and find their way to their respective destinations. The emergence of supporting protocols for new applications and the growth spurt in number of users and the required bandwidth per user results in a very dynamic access environment. A network node must be capable of supporting all the requisite protocols in order to accommodate a variety of source packets.
3. **Privacy and security:** Unlike Frame Relay, ATM networks, and private leased lines, IP networks do not establish a physical or logical circuit between the end users. Consequently, IP networks are prone to a myriad of security attacks, some of which have been well publicized in recent years. Fortunately this issue has received a great deal of attention from the research community, the standardization organizations, and across the industry. A number of IP services can be used to enhance the security and privacy across the Internet. These include *encryption*, *firewalls*, *virtual private networks* and *network address translation*. Virtual private network (VPN) services allow a private network to be configured within a public network. VPN’s rely on some form of tunneling to create an overlay across the IP network by constructing virtual links between the end points of the tunnel. An IP packet is encapsulated in the header of another protocol (e.g., IP) and transported across the network using the forwarding mechanism of that protocol and independently of the address field of the encapsulated packet. A number of IP tunneling protocols have been introduced in recent years including IP/IP, IP/GRE, L2TP [6], IP/IPSec [4] [3] and MPLS [5]. To allow for VPN’s to coexist within the public network, such tunneling protocols must be efficiently supported within the network nodes. Network address translation (NAT) and network address port translation (NAPT) allow for IP level access between hosts in

a (internal) network and the Internet without requiring the hosts to have globally unique IP addresses. They also increase privacy and security as the addresses of the hosts on the internal network are hidden from the public network.

4. **Traffic management:** When users are allowed access at high speeds, it is possible for a limited number of users demanding disproportionate amounts of bandwidth to disrupt service to other users. To ensure that large traffic bursts do not overload small users buffers, and to ensure fairness among different users, traffic policing and shaping must be implemented.
5. **Secure segmentation:** Isolation between users and between groups of users must be provided.
6. **On-Demand Subscription:** On-demand subscription, provisioning, and activation of IP services allows the enterprise customer to respond quickly to unforeseen business needs without costly delays. This requires the dynamic provisioning of guaranteed QoS, privacy and security features across the network and to the enterprise customer's site.
7. **Authentication, Authorization, and Accounting:** Guaranteed, secure, and reliable service requires that the users be authenticated. This implies validating the identity of the users at call setup. Furthermore, authorization is required in order to determine the policy-based services the user can use. Finally the user's connection duration and type of service must be logged for accounting and billing purposes.
8. **Bandwidth and Latency:** Finally, to enable the new services, large bandwidth and low latencies are critical.

In this paper we first describe the traditional approach to provisioning of IP services in the Internet. Next we propose a new architecture for an IP services switch which satisfies all the requirements listed above. To meet these requirements, our proposed architecture is scalable not only in bandwidth, but also in processing power.

2 CPE-Based IP Services

The most common form of IP services deployed today uses customer premises equipment (CPE) devices at the customer site. Figure 1 below shows an example of IP services deployment using CPE devices.

The CPE devices perform customer routing, firewalls, encryption, tunnel generation and termination, as well as network address translation and some form of bandwidth management. While some form of bandwidth management device can be installed at customer premises, there is no end-to-end QoS provisioning. Once the traffic exits the customer's network and enters the WAN, the QoS it receives is undetermined.

Depending on the customer's desired level of control and trust, in-house expertise, and the cost of new applications, hardware, management and upgrades, the SP provides services from pure connectivity to full installation, configuration and management of these services. However, the general trend among the

enterprise customers has been to fully outsource these services. In addition to the fact that some services simply cannot be provided (e.g., QoS, on-demand subscription), the CPE-based approach has several economic and operational drawbacks, some of which are listed below.

1. Expensive up-front capital investment is required that must be shared by the enterprise customer and the service provider depending on their arrangement. In addition, there are significant operational expenses associated with this approach. Deployment of CPE-based services requires a long lead-time. Service preparation involves an initial consultation to identify the customer's needs, ordering and pre-configuring the CPE devices, and shipment. This is then followed by service rollout, which involves an on site visit by technicians to install and integrate the devices. Maintenance and support of these services put a large burden on service providers. They must maintain an inventory of spares and a team of technicians to dispatch to the customer's premises when failures occur or upgrades are needed. For each upgrade or repair the service is temporarily interrupted; a situation that may not be acceptable to the customers since it implies loss of secure connections.
2. Addition of new services and functions requires additional hardware and/or software, which involve additional cost as well as costly staff trips to the site.
3. In many cases, an enterprise has several sites, all of which require enhanced IP services. However, it is often too expensive and cumbersome to deploy these services at every site. Instead, services are deployed at one or two points of access into the public network. The traffic from the other sites is then backhauled to these sites and then onto the Internet resulting in increased traffic on the corporate network.
4. Since no QoS is provisioned across network nodes, network-wide service level agreements cannot be implemented and supported.
5. Since installation, management, and support is not centralized and must be per customer site, it is extremely difficult to scale this approach to a large number of customers.

3 Network-Based IP Services

A new approach to managed IP services is emerging in which these services are moved from the customer premises into the service provider's (SP's) realm of control (Figure 2). In this new paradigm, the intelligence of managed services resides in a reliable broadband aggregation and IP service creation switch, which is deployed at the provider's point of presence (POP) replacing the high-capacity access concentration router. Hence an IP service creation switch is a broadband aggregation router, which is designed to enable value-added services in the public network. As such, it must be highly scalable in speed as well as processing power and it must have carrier-class reliability. To serve a large metropolitan area with a single IP service creation switch, it must be scalable to a large number of interfaces. These interfaces would be needed to handle the multiple sessions that

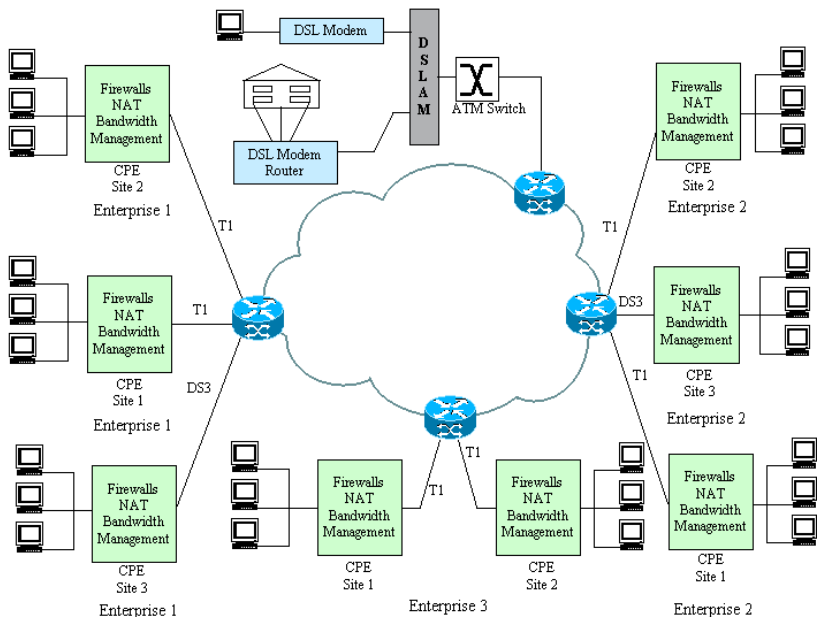


Fig. 1. CPE-based deployment of IP services.

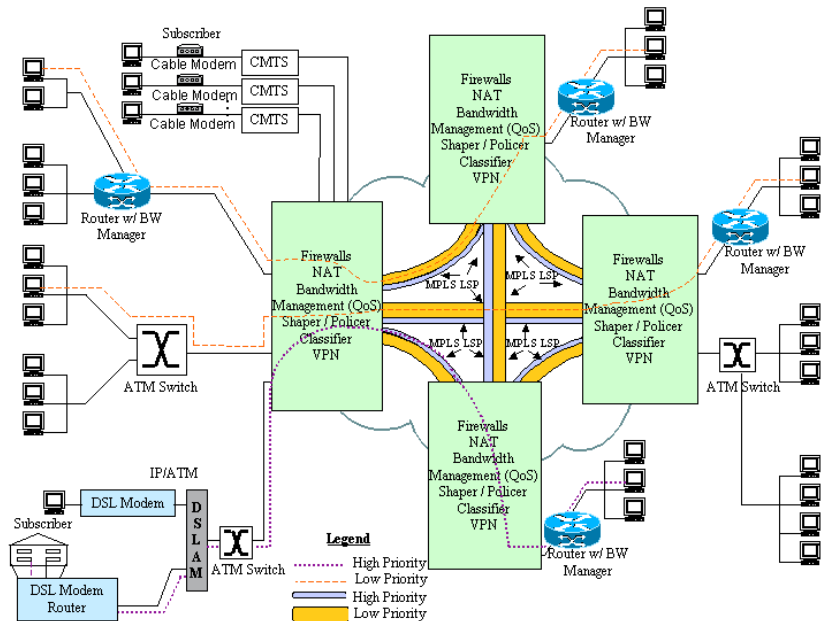


Fig. 2. Network-based deployment of IP services.

can originate in a single household or business as well as those originating from wireless IP enabled devices.

Deployment of IP service platforms provides a great deal of intelligence in the periphery of the network and eliminates the problems associated with the CPE-based approach. In particular,

1. Service provisioning, policy configuration, management, and support can all be provided centrally from the SP's site.
2. Services can be rolled out by the SP at each customer site (rather than at one or two sites) rapidly and cost effectively.
3. The centralized management allows the service providers to monitor performance against their service level agreement (SLA) commitments thereby enabling them to deliver on those commitments.
4. Enterprise customers can initiate and configure new services through a browser interface and obtain detailed network and service performance information through a browser-based interface and verify that their SLA is maintained.
5. The ease of deploying services and the ability to provision end-to-end QoS makes viable new business models such as the Application Service Provider (ASP) model.

In the following section we describe our proposed architecture for such an IP service creation switch.

3.1 Proposed Architecture

Our discussion of the previous sections makes it clear that the IP service creation switches are different from the high bandwidth core routers in that in addition to performing time consuming, repetitive processing for large numbers of routing lookups required of core routers, they must also provide scalable, generalized and flexible computing power which must be easy to program for, among other things, per-user and per-packet processing. Our architectural philosophy maintains a balance between these two types of processing. The need for considerable time-consuming repetitive processing, which has proved to create a bottleneck in the traditional routers, is addressed through specialized hardware. This results in dramatic increases in speed and reductions in delay. On the other hand, the need for flexible, easy to use computing power to enable IP services is addressed through the provision of high-performance general purpose processors which are paralleled and can be scaled to a virtually limitless degree. This architecture is designed to provide scalability in speed/bandwidth, state-space/memory, and processing power.

Figure 3 shows a block diagram of our proposed architecture. Packets enter and exit the switch through media specific physical connection (PHY) cards. The packet entering a PHY card is delivered to the ingress side of its associated line card. After some initial processing, the line card distributes the received packets to a particular Internet Processor Engine (IPE) card through the switch fabric. After performing the necessary processing, the IPE card sends the packet back

through the switch fabric to the egress side of one of the line cards for further processing before allowing the packet to exit the system from the associated PHY card.

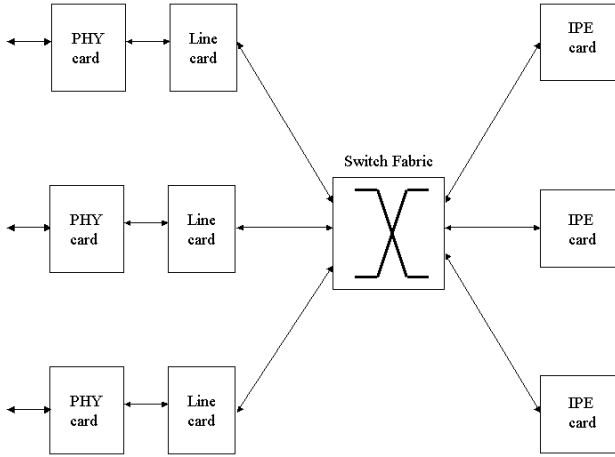


Fig. 3. The system architecture overview.

All the line cards contain identical hardware, but are independently programmable. Similarly all the IPE cards have identical hardware, but are independently programmable. This makes for a simple design and contributes to the scalability of the architecture. If additional processing power is needed, additional IPE cards can be added. Additional users can be supported by adding more line cards and IPE cards.

In general, each line card performs a number of functions. Initially, the line card ingress converts the variable-length input packets into a number of 64-byte fixed-length cells. The stream of cells are examined “on the fly” to obtain important control information including the protocol encapsulation sequence for each packet and those portions of the packet which should be captured for processing. The control information is then used to reassemble the packet and to format it into a limited number of protocol types supported by the IPE cards. Thus, while any given line card can be configured to support packets having a number of protocol layers and protocol encapsulation sequences, the line card is configured to convert these packets into generally non-encapsulated packets of a type that is supported by each of the IPE cards. The line card sends the reassembled and reformatted packet into the switch fabric (in the form of continuous fixed length cells) for delivery to the IPE card that was designated for further processing of that packet.

Although the fixed-length cells which comprise a packet are arranged back to back when sent across the switch fabric, the cells may become interleaved

with other cells destined for the same IPE card during the course of traversing the switch fabric. As a result, the cell stream delivered to an IPE card is in fact an interleaved cell stream. Thus the IPE card will first examine this cell stream “on the fly” (much like the line card) to ascertain important control information. The IPE card then processes this information to perform routing lookup as well as other mid-network processing functions such as policing, packet filtering, PHB scheduling, etc. The control information is also used by the IPE card to reassemble the packet according to the packet’s destination interface. The IPE card then sends the reassembled and reformatted packet back into the switch fabric for delivery to the egress side of one of the line cards (or to one of the IPE cards if the packet requires additional processing).

On the egress side, the line card will again examine the interleaved cell stream and extract the necessary control information. The control information is then used to reassemble and format the packets for their destination interfaces. Additional processing of the outbound packets such as shaping and PHY scheduling is also performed on the line card egress side.

3.2 Data and Control Path

The line cards (on the ingress and egress sides) and the IPE cards host a flexible protocol-processing platform. This platform is comprised of the data path processing and the protocol path processing. The separation of data path processing from protocol processing leads to the separation of memory and compute intensive applications from the flexible protocol processing requirements. A clearly defined interface in the form of dual port memory modules and data structures containing protocol specific information allows the deployment of general-purpose central processing units for the protocol processing units (PPU). These PPU’s support the ever changing requirements of packet forwarding based on multi-layer protocol layers.

Figure 4 illustrates the protocol processing platform. The protocol path processing consists of a number of PPU’s and can be configured for multiple purposes and environments. One of these PPU’s is reserved for maintenance and control purposes and is denoted as Master PPU (MPPU).

The data path processing unit, which is implemented in specialized hardware, consists of the packet inspector, the buffer access controller and the packet manager. This unit extracts, in the packet inspector, all necessary information from the received packets and passes this information on to a selected PPU via the buffer access controller. Furthermore, the packet inspector segments the variable length packet into 64-byte fixed-length cells. The cells are stored in the cell buffer and linked together as linked lists of cells. Once a PPU has selected a packet for transmission it passes the pointer to the packet as well as the necessary formatting and routing information to the data path processing unit. This enables the formatting and the segmenting of the packet. The packet is then forwarded either as a whole (from the line card egress to PHY card) or segmented (from the line card ingress to IPE card and from the IPE card to line card egress) based on the configured interface.

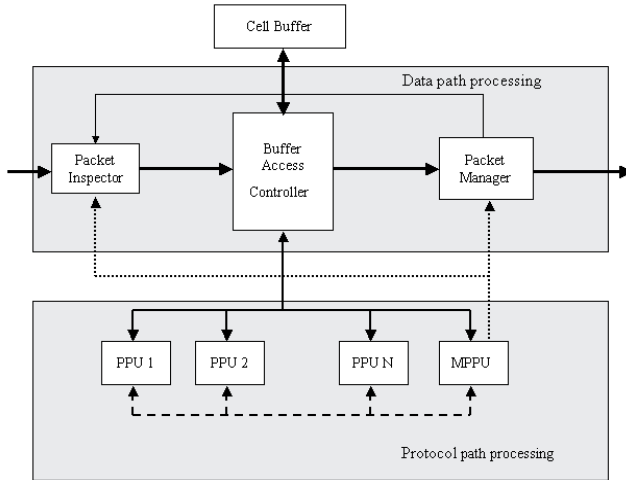


Fig. 4. Protocol processing platform on the line cards and IPE cards.

The set of all ingress users on the system is distributed as evenly as possible across all the IPE cards in the system. This is achieved by the line card which distributes the received packets based on user or tunnel information to a particular IPE card (Figure 5). Within an IPE, the MPPU stores the per-user

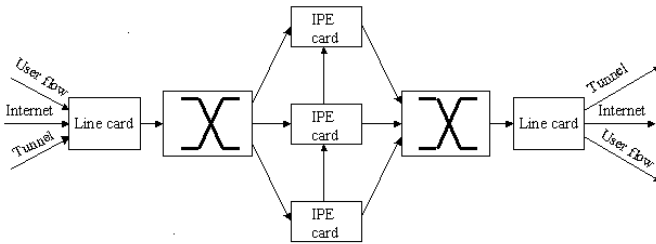


Fig. 5. Load sharing between the IPE cards.

information for the users assigned to that IPE and distributes those users across all the PPU's on that IPE (Figure 6). Each PPU stores a copy of the per-user information assigned to it. Thus each user is associated with one and only one IPE card and one and only one PPU on that IPE. This ensures that packet order is maintained. The procedure of forwarding a packet to a particular IPE card and PPU is denoted as *load sharing*. The key benefits of load sharing are the following.

1. Separation of the data path from the protocol processing path.
2. Incremental provisioning of compute power per packet.
3. Load distribution based on the packet computational needs for a particular user or tunnel.
4. User and tunnel information can be maintained by a single processor thus
 - (a) minimizing the inter-process communication needs,
 - (b) allowing the portability of single processor application software onto the system.

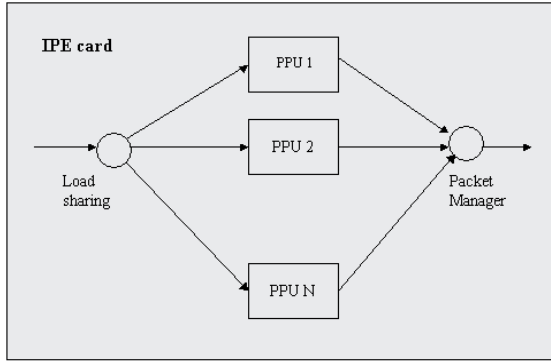


Fig. 6. Processing flow.

3.3 Scheduling and Bandwidth Distribution

As mentioned previously, the set of ingress users are distributed as evenly as possible across all the IPE cards in the system. The forwarding operation is then performed on the IPE cards where each packet is forwarded to an appropriate line card egress (for further transmission to the PHY port) depending on the IP destination address of the packet. Given the bursty nature of the Internet traffic, it is difficult to predict the traffic load from IPE cards to line cards. In particular, over short periods of time, the total traffic from IPE cards destined to a line card may exceed the capacity of the corresponding switch fabric link. In the absence of a scheduling mechanism, the transmitted cells will be dropped by the switch fabric. To prevent this, we have devised a scheduling algorithm which, in response to the requests from line cards and IPE cards, distributes the switch fabric link bandwidths to them. This algorithm is referred to as *bandwidth distribution* in the following.

Bandwidth distribution is performed on one of the MPPUs which is designated as master MPPU. Time is divided into intervals called cycles. At the beginning of each cycle each card transmits its request or demand (determined

by the amount of traffic in its buffers) to the master MPPU. To support differentiated quality of service we assume that traffic consists of P different priority classes and that transmitted requests from the cards are per-priority class. Master MPPU computes a per-priority grant for each card and transmits these grants to the cards. Each card can then transmit traffic of each priority class up to the amount of grant it has received for that class. This scheme resolves contentions within a cycle; in principle, all the cells that are scheduled for a given cycle should be able to reach their destination cards by the end of that cycle. The algorithm is described next.

Let t denote time in terms of the number of cycles and let $d_{ij}(t, p)$, $i, j = 1, 2, \dots, N$, $p = 1, 2, \dots, P$, denote the demand of priority class p from card i for card j , where N is the total number of cards. The $d_{ij}(t, p)$'s are calculated based on the buffer occupancies on each card. At the end of each cycle, each card transmits its demand to the master MPPU. The master MPPU forms P demand matrices $D(t, p) = [d_{ij}(t, p)]$, $p = 1, 2, \dots, P$, from which it calculates P grant matrices $G(t+1, p) = [g_{ij}(t+1, p)]$, where $g_{ij}(t, p)$ denotes the bandwidth grant for traffic class p assigned to card i and destined to card j during the cycle t .

Grants are allocated based on strict order of priority. The bandwidth distribution algorithm tries to fulfill the demands of the highest priority class first. If any additional bandwidth is left on any of the input or output links, it is assigned to the next priority class according to their demands. In the following we describe an algorithm for computing the grant matrices. To simplify notation we remove the time and priority indices of the request and grant matrices. Furthermore each switch fabric port is assumed to have a capacity of C cells/sec.

Bandwidth distribution algorithm

Step 0

Given the demand matrix D , calculate the total demand of every row and column denoted by $u_i^{(0)} = \sum_j d_{ij}$ and $v_j^{(0)} = \sum_i d_{ij}$, respectively. Calculate a_0 , where

$$a_0 = \min\left\{\frac{C}{u_i^{(0)}}, \frac{C}{v_i^{(0)}}, i = 1, 2, \dots\right\}. \quad (1)$$

Each input port can be assigned a bandwidth of a_0 for each output port without violating the capacity constraint of any input or output link. Suppose the minimum is achieved for row k . Then $a_0 = C/u_k^{(0)}$. For $j = 1, 2, \dots, N$, let $g_{kj} = a_0 d_{kj}$. Row k (input link k) has reached its maximum capacity now. It will be eliminated from further consideration. The situation is similar if the minimum is achieved for a column. All other input/output demands will receive the same amount of bandwidth at this stage.

Step l , $1 \leq l \leq 2N - 1$

Up to this point l row or columns have been eliminated. Calculate the unfulfilled demands for every row and column. If a row, say row m , was eliminated in step $l - 1$, then $u_i^{(l)} = u_i^{(l-1)}$ and $v_i^{(l)} = v_i^{(l-1)} - d_{mi}$, $i = 1, 2, \dots, N$. Similarly, if a column, say column n , was eliminated in step $l - 1$, then $u_i^{(l)} = u_i^{(l-1)} - d_{in}$ and

$v_i^{(l)} = v_i^{(l-1)}$, $i = 1, 2, \dots, N$. Calculate the residual bandwidth on each row and column

$$r_i^{(l)} = r_i^{(l-1)} - a_{l-1} u_i^{(l-1)}, \quad s_i^{(l)} = s_i^{(l-1)} - a_{l-1} v_i^{(l-1)},$$

where $r_i^{(0)} = s_i^{(0)} = C$ for all i . Now evaluate the maximum bandwidth increment,

$$a_l = \min \left\{ \frac{r_i^{(l)}}{u_i^{(l)}}, \frac{s_i^{(l)}}{v_i^{(l)}}, i = 1, 2, \dots, \right\} \quad (2)$$

and allocate bandwidth $\sum_{t=0}^l a_t$ proportionally to all the requests of the row or column that achieves the minimum, i.e., if the minimum is achieved for, say, row r , then $g_{rj} = (\sum_{t=0}^l a_t) d_{rj}$ for all j .

If only one row or column is left, we allocate all the residual capacity to the requests in that row or column keeping in mind the capacity constraint of the switch fabric input and output ports.

The above algorithm has the following interesting properties.

1. Except for the last step of the algorithm, all input and output ports corresponding to the rows and columns which are eliminated have 100% utilization.
2. The algorithm is fair in the following sense. Except for the input/output pairs which receive bandwidth allocation in the last step, in every step of the algorithm the grant allocated to the input/output pairs is proportional to their requests. This property does not hold for the last row or column since in that case all the residual capacity is assigned to the remaining requests.
3. For an $N \times N$ matrix, there are at least N and at most $2N$ iterations. It can be shown that this algorithm requires $O(N^2)$ additions and multiplications. This is the lowest complexity that any centralized algorithm can achieve.

Credits and grants

If grants are generated in direct response to demands from cards, then each cell must undergo a latency of at least one cycle. In general the length of a cycle (on the order of msec.) is too large compared to the delay tolerance of real-time traffic. In order to eliminate this latency we issue every card a fixed credit line of L_p cells for the priority class p . A card that has no grants can transmit up to L_p cells from priority class p without having obtained any grants for it. The algorithm is described in the following.

Consider priority class p and cycle t . Let $B_{ij}(t, p)$ and $c_{ij}(t, p)$ denote the buffer occupancy and the credit of card i for card j at the end of cycle t , respectively. Also let $T_{ij}(t, p)$ and $A_{ij}(t, p)$ denote the number of cells transmitted and the number of cells that arrived, respectively, during this cycle. Then the buffer occupancy of card i for card j at the end of cycle t is given by

$$B_{ij}(t, p) = \max 0, B_{ij}(t-1, p) - T_{ij}(t, p) + A_{ij}(t, p). \quad (3)$$

Furthermore, the remaining credit at the end of cycle t is given by

$$c_{ij}(t, p) = \min \{ L_p, c_{ij}(t-1, p) - T_{ij}(t, p) + g_{ij}(t, p) \}. \quad (4)$$

The number of cells that can be transmitted during a cycle t is then given by

$$T_{ij}(t, p) = \min\{c_{ij}(t - 1, p), B_{ij}(t - 1, p)\}. \quad (5)$$

The new demand is now calculated such that if granted, the card will have a credit equal to L_p plus the traffic remaining in its queue from the previous cycle, i.e.,

$$d_{ij}(t, p) = L_p + B_{ij}(t, p) - c_{ij}(t, p). \quad (6)$$

The initial conditions are given by

$$d_{ij}(0, p) = B_{ij}(0, p), \quad c_{ij}(0, p) = L_p.$$

Having obtained its grant $g_{ij}(t, p)$ from the master MPPU, card i computes its new credit for the next cycle. It then transmits traffic up to the level of its credit from its buffers.

We have performed a set of simulations in order to measure throughput and delay in different parts of the system and to evaluate the effectiveness of our approach in reducing the switch fabric contention. Our results show that this approach achieves nearly 100% throughput and can easily support differentiated quality of service in terms latency for different classes of traffic [2].

4 Conclusion

In this article, we present an architecture that meets the stringent requirements of an IP service creation switch. The proposed architecture enables the support of user management, tunnel management, logical link management, per-user policing, per-user traffic shaping, per-user QoS control with Diffserv support, per-user buffer management, per-user packet classification, packet filtering, NAT, and management database support, all for a very large number of users. The simultaneous control plane support of the requisite protocols such as the routing protocols, the tunneling protocols and the QoS protocols makes this a very well-matched architecture for a scalable IP services switch.

References

1. Blake, S., et. al., "An Architecture for Differentiated Services," RFC 2475, December 1998.
2. Hegde, Manju and Mort Naraghi-Pour, "Scaling High-Speed Shared-Memory Switch Fabric Buffers," to appear in the *Proceedings of the IEEE Workshop on High Performance Switching and Routing*, May 2001, Dallas, TX.
3. Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, November 1998.
4. Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP," RFC 2407, November 1998.
5. E. Rosen, A. Viswanathan, R. Callon, "Multi-protocol Label Switching Architecture," August 1999, IETF draft-ietf-mpls-arch-06.txt.
6. Townsley, W., et. al., "Layer Two Tunneling Protocol "L2TP"," Internet-Draft, Feb. 1999.

On Implementation of Logical Time in Distributed Systems Operating over a Wireless IP Network

Denis A. Khotimsky¹ and Igor A. Zhuklinets²

¹ Bell Labs, Lucent Technologies, Holmdel NJ 07733, USA
dkhotimsky@lucent.com

² Mozhaysky University, St. Petersburg, Russia
jkl@solaris.ru

Abstract. We address the problem of maintaining logical time in a distributed system operating over a special type of a network: one whose underlying graph is represented by fully-connected mesh of star-topology subgraphs. Such a graph is generated, for example, by a wireless IP network in which mobility service agents, interconnected by an IP backbone, provide wireless connectivity to mobile hosts currently present in the associated location areas. We discuss the complementary logical clocks for the wireline and wireless segment of the network and their integration into an isomorphic logical time system.

1 Introduction

Tracking causal relations between activities occurring in distinct physical locations plays a critical role in monitoring, analyzing and debugging the behavior of distributed systems. Yet understanding of a distributed execution, detecting causality or concurrency (*i.e.*, causal independence) between local events and accounting for the corresponding non-determinism presents a formidable engineering challenge. The task becomes notoriously difficult as the complexity of the underlying communication network and the number of potentially interacting sites grows.

Logical time provides a standard tool to characterize causality in a distributed system. A logical clock protocol, that runs on top of the regular message passing communication, assigns a *timestamp* to each event. Comparing the timestamps allows to draw conclusions (deterministically, with quantitative probability, or in some likelihood) about the existence of causal relation between the events. The simplest causal relation, the relation *happened-before*, which has been defined for the processes with the totally ordered events [1], requires *vector clock* to characterize causality in an isomorphic fashion [2,3]. This mechanism employs event timestamps and message tags represented by integer vectors whose size is equal to the number of processes in the system [4]. For many applications, such a cost appears intolerably high. As a response to the overhead problem, the research activities have focused on efficient encoding schemes [5], the trade-offs

between online and offline communications [6,7], and the trade-offs between clock accuracy and complexity [8,9]. In a parallel development, a generalized approach to causality has been sought with the objective to extend the methodology to the processes with the partially ordered events [10,11] as well as to the special types of processes and underlying networks [12,13].

In the present work, we focus on the isomorphic tracking of causality in a large distributed system running over an IP network with the wireless mobile access segment. In such a network, a mobility service agent (MSA) provides wireless access over a radio or infrared links to a group of mobile users (nodes) in its location area, or subnet. The MSA's themselves are interconnected by a fixed wireline network. A mobile node can communicate with the other nodes, both mobile and static, exclusively via the MSA of its current location area. When a node moves from one location area to another, the link with the first MSA is brought down and a new link with the MSA of the second location area is established. The model is applicable to the advanced wide-area wireless systems, as well as to corporate wireless LANs. Depending on the actual operating environment, the MSA function can be represented by a base station, a wireless router, a radio network controller, a packet radio support node, or a packet data service node.

In a distributed system running over such a network, an MSA with the mobile nodes logically linked to it may be viewed as an abstracted process with a partially ordered event set. The overall task of characterizing causality can be partitioned into the smaller tasks applied to the set of abstract processes, on one hand, and a set of the mobile nodes of the same location area, on the other hand. According to this partition, we define a system model and introduce a refined definition of the causal precedence relation for a distributed system containing mobile nodes interacting via mobility service agents, develop two separate logical clocks for the wireline and wireless segments of the network, and show how they interact to form an integrated isomorphic logical time system.

2 System Model

Fixed and Mobile Network Segments

We consider a distributed system containing interacting processes running at *mobile nodes* (MN) and *mobility service agents* (MSA), as shown in Fig. 1. When an MN moves into a location area associated with a specific IP subnet, it uses a registration mechanism to establish its presence with the MSA of this subnet and to obtain a temporary care-of IP address [14]. All communications within a subnet are carried over a set of radio or infrared links between the MSA and mobile nodes. The MSA, which is connected to a routing node of an underlying IP network of arbitrary topology (shown with the dash lines in the Figure), handles all incoming and outgoing traffic of a mobile node. Specific network topology as well as the care-of address assignment and tunneling mechanisms remain fully transparent to the mobile applications. The set of mobility service agents is modeled as a vertex set of a fully connected graph. Each edge of the

graph represents a full-duplex, non-FIFO channel with a non-zero probability of a message loss (solid lines).

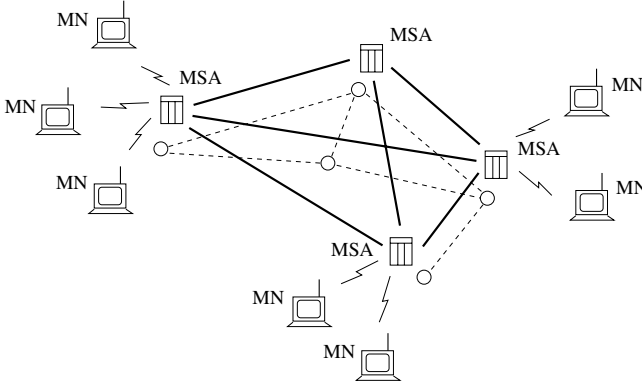


Fig. 1. Distributed system running over a mobile IP network: MN - mobile node, MSA - stationary mobility service agent.

Each subnet containing a mobility service agent and a number of mobile nodes is modeled as a star-topology graph, where each edge corresponds to an error-free full-duplex FIFO channel with a non-zero probability of message loss in either direction. The latter property reflects possible link impairment due to landscape features or foreign objects moving into the signal propagation path.

In order to transmit and receive messages, a mobile node has to be registered with an MSA. At most one registration can be valid at any given time. When a mobile node moves from one subnet to another, its link to the mobility agent of the first subnet is brought down, and a new link with the mobility agent of the second subnet is established. In the course of such transition, a mobile node may have to obtain a new care-of IP address. The details of this mechanism including the signaling message exchange also remain transparent to the applications.

Event Types and Causality Relations

A *process* in a distributed system is the program code running either at a mobile node or a mobility service agent. Any change in a process state constitutes an *event*. Without loss of generality, we further consider only *communication events*, i.e., ones associated with sending or receiving an information message. Events occurring at any given process (MN or MSA) are *proper events* of that process. Proper events of any process are sequentially ordered. An example of a distributed system with communication events is shown in Fig. 2.

An event occurring at an MN is either a message send event (NSEND) or a message receive event (NRECV). *Registration* represents a special type of event that involves an exchange of signaling messages between the node and mobility

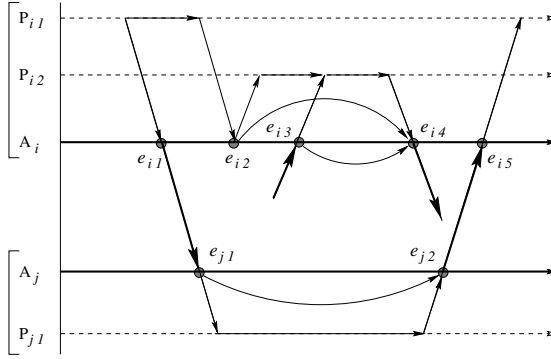


Fig. 2. An example of communication in a system with three mobile nodes and two mobility service agents.

service agents. Events occurring at an MSA are classified as either MSA-IN, MSA-OUT, or MSA-RELAY. An MSA-IN event is atomically composed of receiving a message over the fixed part of the network and forwarding it to an MN in the location area of the given MSA. Similarly, an MSA-OUT event is composed of receiving a message from an MN over a wireless link and forwarding it over the fixed network, whereas an MSA-RELAY event is a combination of receiving a message from one MN and forwarding it to another MN in the agent's location area.

Definition 1 (Elementary internal causality) *For a given mobility service agent A , elementary internal causal relation exists between events e and f , if one of the following conditions holds:*

- events e and f belong the same mobile node P , e precedes f in the ordered sequence of P 's events, and P is registered with A at occurrence of event e as well as at occurrence of event f ;
- event e is a NSEND event occurring at a mobile node while this node is registered with A and f is the corresponding MSA-OUT or MSA-RELAY event of A ;
- event e is an MSA-IN or MSA-RELAY event of A and f is the corresponding NRECV of a mobile node occurring while this node is registered with A .

Definition 2 (Internal causality) *The internal causal relation is the transitive closure of the elementary internal causal relation. The internal causality with respect to a given mobility service agent A between events e and f is denoted $e \xrightarrow{A} f$.*

Definition 3 (External causality) *The external causal relation exists between events e and f , if e is a MSA-OUT event occurring at one MSA and f is the corresponding MSA-IN event occurring at another MSA.*

Definition 4 (Causal precedence) *The causal precedence relation on the set E of events in a distributed system is a transitive closure of the union of internal*

and external causal relations. The causal precedence between events e and f is denoted $e \rightsquigarrow f$.

In Fig. 2, the elementary internal causality is shown with thin straight arrows, arches denote the internal causality, and thick arrows mark the external causality.

3 Logical Time Protocol

Wireline Segment

Consider a distributed system formed by interacting processes that correspond to the set of the mobility service agents and assume that the respective internal causal relations are known *a priori* (before the execution of the logical time protocol). Such an abstraction results in a general process model with the partially ordered event sets. The needs of that model are conventionally met by the bit-matrix (BM) logical time system [10].

For each event in the system containing N processes, the BM clock assigns a timestamp of N components. Each component itself is a vector of bits, size of which (ignoring the trailing zeros) varies with the number of events in the corresponding process. The process initialization event receives a timestamp containing all-zero vector components. On occurrence of event $e_{k,i}$, *i.e.*, i -th event in process A_k , its BM timestamp is found by computing bit-wise OR over the timestamps of all events $e_{k,j}$, such that $e_{k,j} \xrightarrow{A_k} e_{k,i}$; in addition the i -th bit of the proper bit-vector is set to 1. Each message in the system carries the time tag equal to the timestamp assigned to the message send event. On receipt of a message, the BM timestamp of the receive event $e_{k,i}$ is computed as a bit-wise OR over the timestamps of all internally precedent events $e_{k,j} \xrightarrow{A_k} e_{k,i}$ as well as the timetag of the message; in addition the i -th bit of the proper component is set to 1. Timestamp $T(e)$ of any event e , assigned by the BM clock, is *complete* with respect to the causal past $Past(e)$, since $a \rightsquigarrow e$ implies $T(e)|_a = 1$. Furthermore, as the opposite is true as well, the BM timestamp $T(e)$ is *indicative*: $a \in Past(e) \iff T(e)|_a = 1$.

The substantial disadvantage of the BM clock lies in its communication complexity. Two approaches increasing the feasibility of the method have been considered in [11]. The idea of the *dependency sequence* timestamp compression technique is to reduce the overhead on average, while maintaining the essential information content of the BM timestamps. The two-tier *hierarchical clock* transmits less information on-line, but provides for the restoration of indicative event timestamps by means of *off-line query exchange*. The efficiency of the latter algorithm depends on the presumed knowledge by the MSA of all causal dependencies between events of the registered mobile nodes, including those dependencies whose causality chains contain external causal links. If the MSA's *a priori* knowledge is restricted to the internal causal relations, as defined above, the number of the off-line queries required by the two-tier hierarchical clock for the indicative timestamp restoration in the worst case reaches the order of transmitted information messages [15].

An efficient alternative to the bit-matrix clock is offered by the compact bit-matrix clock (CBM) mechanism. The underlying idea of CBM is to minimize the weight (*i.e.*, the number of bits set to one) of the timestamps assigned on-line, while retaining the possibility to restore the indicative timestamp of any event off-line using not more than N queries, where N is the number of processes in the system. The sparse bit-matrices used to tag the messages in CBM allow more efficient compression (with any known lossless technique, of which the dependency sequences is one example) and hence incur significantly smaller on-line communication overhead. The theory of compact bit-matrix clock is based on the following result. Let *local causal past* $Past_i(e)$ be the subset of $Past(e)$ containing all events that occur at the abstracted process A_i .

Proposition 1 *For any given event, the off-line restoration of an indicative timestamp requires not more than N queries, if for any event e occurring at process A_i , the timestamp $T(e)$ assigned on-line*

- *is a valid partial timestamp;*
- *is complete with respect to $Past_i(e)$, and*
- *for each $j \neq i$, is complete with respect to maximal set¹ $Max(Past_j(e), \sim_j^{A_j})$.*

In CBM, a message tag does not coincide with the timestamp of the corresponding message send event. In its basic form, a CBM message tag carries the proper bit-vector that contains just a single bit referring to the send event itself. In addition, maintaining an *absorption matrix* allows to exclude from message tags any redundant causal past references that can be inferred by the recipient.

Wireless Segment

The logical time protocol proposed for the wireless segment of the network adapts the well-known isomorphic vector clock to the wireless environment. First, since the membership in the group of mobile nodes within the service area of a given MSA changes over time, maintaining a logical time vector at each mobile node may not be easily achievable. Second, since mobile nodes conduct their communication via the mobility service agent, it is natural to delegate handling of the vector time to the MSA, thus reducing the overhead on the wireless links, where bandwidth is a critical resource. Third, since the wireless link is, by definition, lossy, the means to account for possible undelivered messages in either direction should be incorporated into the protocol.

The centralized management of the vector time employs the MSA events as representative images of the mobile node events. Accordingly, the vector timestamps assigned to MSA events may reflect the causal relations between the corresponding mobile node events. We refer to this type of causal relations between events of the same MSA as *induced causality*. There are two types of induced causal relations:

¹ By definition, the *maximal set* of S with respect to an order relation $\prec$, $Max(S, \prec)$, contains all $x \in S$, such that no other element in S is strictly superior to x .

- an MSA-OUT (MSA-RELAY) event causally precedes a subsequent MSA-OUT (MSA-RELAY) as long as both associated messages originate at the same mobile node registered with the given MSA;
- an acknowledged MSA-IN (MSA-RELAY) event causally precedes a subsequent MSA-IN (MSA-RELAY) event as long as both associated messages are destined to the same mobile node registered with the given MSA and both messages are successfully received by that node.

As a message transmitted by an MSA to the mobile node may be lost while in transit, the induced causality of the latter type can not be established at the time of the second MSA-IN event occurrence. Instead it is detected upon occurrence of a subsequent MSA-OUT event that acknowledges the receipt of the message. The wireless segment protocol accounts for the induced causality in order to simplify process of determining causal precedence between mobile events.

The wireless segment protocol is outlined in Fig. 3, 4. Class `Rmsg_t` refers to the wireless message overhead, which contains the following fields: `locid` is the sender's event id, `baseid` is the id of the most recent acknowledged message from the MSA to the mobile node, and bit-vector `list[]` is the indicator of the set of MSA messages received by the mobile node. In Fig 4, bit-vector `Lpast[]` is the indicator of the local past maximal set, extracted from the received external message.

Hand-Off Procedure

When a mobile node moves from one location area to another or powers up after changing location, the *hand-off* procedure is executed. In addition to the new IP address registration, the hand-off requires an exchange of signaling messages between the MSA's of two location areas.

First assume that the departing MN is able to inform its old MSA of a pending hand-off. Then the notification message carries the overhead fields of a regular HSEND event. The MSA handles its receipt as a MSA-OUT event, establishing the internal causal relations, forming the compact bit-matrix tag, and forwarding the hand-off request to the mobility agent with which the node is going to register. In addition to the CBM tag, the request carries the MN's event id. When the new MSA receives the request, it uses the CBM tag as a reference into its external history to compute the local vector time. It then assigns a new local node id and performs initialization: `msgidvect` component is set equal to the event id passed with the request, whereas `basemsgvect` component is set to 0, indicating the hand-off event itself. No protocol information needs to be sent by the MSA to the mobile node, as any message received later from the MN acknowledges the hand-off by default.

Alternatively, the hand-off can be executed without prior notification of the old MSA. Upon receipt of a message from a previously unregistered node, the MSA of a new location area puts the message on hold while forwarding the request with original message tag to the MN's old MSA. The old MSA processes the request as if it was a local MSA-OUT event (see above), forms a CBM tag

```

class Node_t is          // mobile node object

    int   locevid;        // local event id
    int   basemsgid;      // last successfully acknowledged message
    bit   msglist[ ];     // list of received messages
    int   lastmsg;        // last received message

    void   HRECV (Rmsg_t msgin)
    var
        int   baseadv, msgloss;
    begin
        locevid := locevid + 1;
        baseadv := msgin.baseid - basemsgid;
        if (baseadv > 0) then
            truncate(msglist, baseadv);
            basemsgid := msgin.baseid;
        end
        msgloss := msgin.locid - lastmsg - 1;
        msglist := msglist || '0' * msgloss || '1';
        lastmsg := msgin.locid;
    end

    Rmsg_t   HSEND ( )
    begin
        Rmsg_t   msgout;
        locevid := locevid + 1;
        msgout.locid := locevid;
        msgout.baseid := basemsgid;
        msgout.list := msglist;
        return msgout;
    end
end

```

Fig. 3. Mobile node behavior in the wireless segment logical time protocol

and forwards the hand-off confirmation to the new MSA. The latter initializes the data structures and transmits the messages that had been earlier put on hold to their destinations.

Establishing Causal Relation

To find out the causal relation between two given mobile nodes events e and f , first check whether the mobile nodes at time of the event occurrence were registered with one and the same MSA. If this is the case, the local vector time, maintained by the MSA, completely characterizes causality. Otherwise, assume that $e \rightsquigarrow f$ is the hypothesis is to be tested, and event e occurs at a mobile node registered with MSA A_i , whereas event f occurs at a mobile node registered with MSA A_j . For the base message send event of event f , as well as all events referred to by its list of received but unacknowledged messages, find the union of their $Max(Past_j(\cdot), \rightsquigarrow^{A_j})$ sets, as determined by the respective CBM timestamps. Then query MSA A_j with respect that set and event e . To answer the query, MSA A_j finds the earliest MSA-OUT or MSA-RELAY event that is aware of event e , and determines whether this event belongs to the causal past of any of

```

class MSA_t is                                     // mobility service agent object

    int  msgidvect[ ];                             // message id per mobile node
    int  basemsgvect[ ];                           // last acknowledged message per node
    vtime.t  ExtHist[ ];                           // external send history
    vtime.t  IntHist[ ][ ];                         // internal send history

    Rmsg.t  MSA-IN (int nodeid, bitvect Lpast[ ])
    var
        Rmsg.t  msg;
        vtime.t  locvtime;
    begin
        locvtime := max{e∈Lpast[ ]} ExtHist[e];
        msgidvect[nodeid] := msgidvect[nodeid] + 1;
        msg.locid := msgidvect[nodeid];
        msg.basemsgid := basemsgvect[nodeid];
        IntHist[nodeid][msgidvect[nodeid]] := locvtime;
        return msg;
    end

    void  MSA-OUT (int nodeid, int xev, Rmsg.t msg)
    var
        vtime.t  locvtime;
    begin
        locvtime := max{i≥0, msg.list[i] = 1}
                        IntHist[nodeid][msg.baseid + i];
        basemsgvect[nodeid] := msg.baseid + length(msg.list);
        IntHist[nodeid][basemsgvect[nodeid]] := locvtime;
        locvtime[nodeid] := msg.locid;
        ExtHist[xev] := locvtime;
    end
end

```

Fig. 4. Mobility agent behavior in the wireless segment logical time protocol

the query events. If so, the hypothesis $e \rightsquigarrow f$ is confirmed, otherwise, it is known to be false.

4 Conclusions

In the present work, we have focused on the isomorphic tracking of causality in a large distributed system running over an IP network with the wireless mobile access. In such a network, mobile nodes communicate with each other and with the rest of the network via the mobility service agents. The network can be naturally partitioned into wireline and wireless segments with the complementary logical time protocols running on each segment. We have used the Compact Bit-Matrix clock mechanism in the wireline segment and have proposed a centralized modification of the vector time protocol to manage the logical time in the wireless segment of the network. We have demonstrated how the two mechanisms can be integrated into an isomorphic logical time system. The integrated protocol can be readily used to handle hand-offs and requires a single query to test a causality hypothesis between any pair of mobile node events.

References

1. Leslie Lamport. Time, clocks, and the orderings of events in a distributed system. *Communications of ACM*, 21:558–564, 1978.
2. Colin J. Fidge. Timestamps in message-passing systems that preserve the partial ordering. In *Proc. 11th Australian Comp. Sci. Conf.*, pages 56–66, 1988.
3. Friedemann Mattern. Virtual time and global states of distributed systems. In M. Cosnard et al., editors, *Proceedings of the International Workshop on Parallel and Distributed Algorithms*, pages 215–226, Amsterdam, 1989. Elsevier Science Publishers.
4. Bernadette Charron-Bost. Concerning the size of logical clocks in distributed systems. *Information Processing Letters*, 39(1):11–16, September 1991.
5. Mukesh Singhal and Ajay Kshemkalyani. An efficient implementation of vector clocks. *Information Processing Letters*, 43(1):47–52, August 1992.
6. J. Fowler and W. Zwaenepoel. Causal distributed breakpoints. In *Proc. 10th Intl. Conf. Distributed Computing Systems (ICDCS-10)*, Paris, France, May-June 1990.
7. Toshimitsu Masuzawa and Nobuki Tokura. A causal distributed breakpoint algorithm for distributed debugger. In *Proceedings of ICICE Fall Conference (SD-1-8)*, pages 373–374, 1992.
8. Francisco J. Torres-Rojas and Mustaque Ahamad. Plausible clocks: Constant size logical clocks for distributed systems. In Özalp Babaoglu and Keith Marzullo, editors, *Distributed Algorithms, 10th International Workshop, WDAG '96*, volume 1151 of *Lecture Notes in Computer Science*, pages 71–88, Bologna, Italy, October 1996. Springer Verlag.
9. Denis A. Khotimsky and Igor A. Zhuklinets. Hierarchical vector clock: Scalable plausible clock for detecting causality in large distributed systems. In *ICATM'99. Proceedings of the 2nd International Conference on ATM*, pages 156–163, Colmar, France, June 1999.
10. J. Ahuja, T. Carlson, and A. Gahlot. Passive-space and time view: Vector clocks for achieving higher performance, program correction, and distributed computing. *IEEE Transactions on Software Engineering*, 19(9):845–855, September 1993.
11. Ravi Prakash and Mukesh Singhal. Dependency sequences and hierarchical clocks: Efficient alternatives to vector clock for mobile computing systems. *Wireless Networks*, 3(5):349–360, 1997.
12. Colin J. Fidge. Logical time in distributed computing systems. *Computer*, 24(8):28–33, 1991.
13. Koenraad Audenaert. Clock trees: Logical clocks for programs with nested parallelism. *IEEE Transactions on Software Engineering*, 23:646–658, October 1997.
14. Charles E. Perkins, Editor. IP Mobility Support. *Request for comments, RFC 2002*. October 1996.
15. Denis A. Khotimsky and Igor A. Zhuklinets. Compact bit-matrix clock with $O(n)$ complexity of the off-line timestamp restoration. *Unpublished manuscript*, 2000.

Performance of an Inter-segment Handover Protocol in an IP-Based Terrestrial/Satellite Mobile Communications Network

L. Fan, M.E. Woodward, and J.G. Gardiner

Department of Computing, University of Bradford, Bradford, BD7 1DP, United Kingdom
{L.Fan, M.E.Woodward, [J.G.Gardiner](mailto:J.G.Gardiner@Bradford.ac.uk)}@Bradford.ac.uk

Abstract. The convergence of the mobile communications and Internet technologies has become a major driving force in the next generation telecommunications world. In order to realise the dream of global personal communications and allow the users to access information anywhere, at any time, the integration of terrestrial and satellite communications networks becomes necessary. Because the inter-segment handover is regarded as a key element to implement the seamless integration of terrestrial/satellite communications networks, it has been placed among top issues in the research of global personal communications. In this paper, by adopting mobile-IP, intelligent network and dual-mode mobile terminal technologies, the architecture of an IP-based integrated terrestrial/satellite mobile communications network is proposed on the infusion of mobile communications and Internet technologies. Based on this architecture, an inter-segment handover protocol is described, and its performance is analysed.

1 Introduction

In our information society, the last decade has been marked by the tremendous success of mobile communications and the Internet. Now, the convergence of the mobile communications and Internet technologies has become a major driving force in the next generation telecommunications world and leads to the birth of IP-based mobile communications networks, the essence of which is to allow the users to access information anywhere, at any time. It is generally agreed that the terrestrial and satellite communications networks should work together to realise global personal communications. Because the inter-segment handover is regarded as a key element to implement the seamless integration of terrestrial/satellite communications networks, it has been placed among top issues in the research of global personal communications. In this paper, by adopting mobile-IP, intelligent network and dual-mode mobile terminal technologies, the architecture of an IP-based integrated terrestrial/satellite mobile communications network is proposed on the infusion of mobile communications and Internet technologies. Based on this architecture, a functional model is built and an inter-segment handover protocol is described, and its performance is analysed using both a mixture of simulation and analytical modelling.

2 Network Architecture

As shown in Figure 1, the components of proposed network architecture are designed to be consistent with the next generation mobile network architecture defined by the Third Generation Partnership Project (3GPP) [1] and ITU [2]. This architecture is divided into three subnetworks: mobile equipment, access network and core network domains.

Mobile equipment domain – This consists of the mobile user and the mobile terminal. The mobile user is a mobile personal computer that can run a number of application programs, and provides real-time and non real-time services.

The mobile terminal is considered to be a dual-mode terminal that can connect to both terrestrial and satellite access networks, and can perform a variety of functionalities. The mobile user and mobile terminal can be separated to allow the mobile user to connect to any mobile terminal.

Access network domain – There are two types of access networks: the terrestrial access network, which consists of Base Transceiver Station (BTS) and Radio Network Controller (RNC), and the satellite access network includes satellites and Fixed Earth Stations (FES).

Core network domain – The IP-based core network is a part of Internet and comprises the gateways, the intelligent networks and the home agent.

The gateway is a router on the mobile node's visited network and provides routing services to the mobile user while the mobile user attaches to it. It also provides the switch function and is fully controlled by the intelligent network.

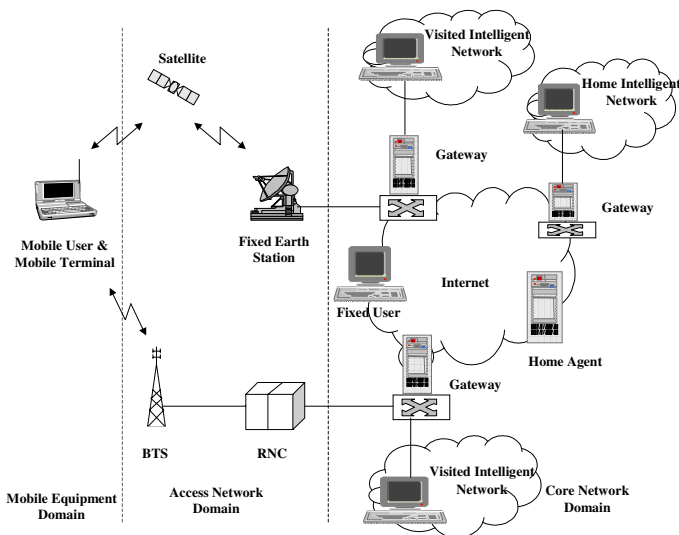


Fig. 1. The Network Architecture

The intelligent network is the “intelligent part” of the mobile communications networks, and takes charge of call and connection control, mobility management, and service management. There are three intelligent networks in the core network: one is

the home intelligent network, where the mobile terminal registers, and the other two are visited intelligent networks, where the mobile terminal may visit. These two visited intelligent networks take charge of terrestrial and satellite mobile networks separately. When the mobile terminal roams to a visited network, the visited intelligent network will exchange relative information with the mobile terminal's home intelligent network.

The home agent is a router on a mobile user's home network and maintains subscriber information and tunnels datagrams to the mobile user when it is away from its home network [3].

3 Functional Model

Figure 2 shows the functional model of the proposed network architecture.

The functional entities in this model are consistent with the definition of ITU [4], and their functionalities can be deduced from their names. This model can be divided into five parts: the MU, the MT, the RNC/FES, the intelligent networks and the fixed user.

Both MU and Fixed User have only one functionality: User Identification Management Function (UIMF), which handles the identification and service related to the user.

In the MT there are four associated functionalities: Call Control Agent Function (CCAF), Mobile Control Function (MCF), Radio Access Control Agent Function (RACAF), and Mobile Radio Transmission and Reception (MRTR). CCAF takes charges of call control, and MCF handles service control. RACAF and MRTR takes charge of radio access control related functions.

In the RNC and FES, there are two associated functionalities: Radio Access Control Function (RACF) and Radio Frequency Transmission and Reception (RFTR). These two entities take charge of radio access control related functions.

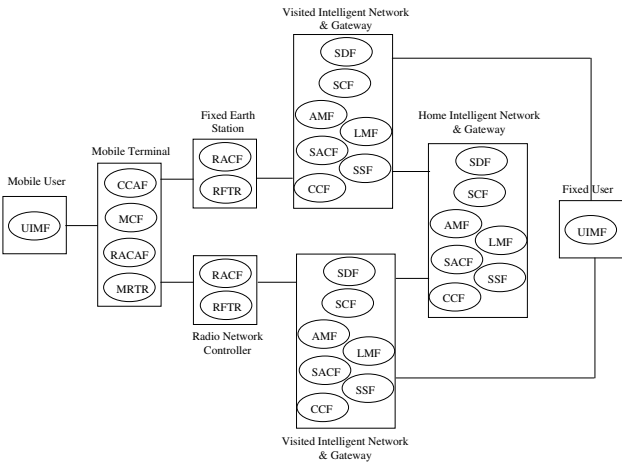


Fig. 2. Functional Model

There are seven associated functionalities in the intelligent network: Service Data Function (SDF), Service Control Function (SCF), Authentication Management (AMF), Location Management (LMF), Service Access Control Function (SACF), Call Control Function (CCF) and Service Switching Function (SSF). SDF and AMF controls the storage and provide data-related services. SCF manages all the service control and mobility management in the mobile communications network. CCF controls the call/connection processing. SACF provides both call-related and call-unrelated processing and control. SSF maintains the interaction between SCF and CCF. LMF takes charges of terminal mobility and location management.

The Generic Radio Access Network (GRAN) concept is used in this paper to divide the access network's functionalities into radio-independent and radio-dependent parts, which allows different radio access modules to be connected to the same network infrastructure via a unique interface.

Since the proposed inter-segment handover protocol in this paper are radio-independent, it can be used in both terrestrial and satellite systems.

4 Inter-segment Handover Protocol

Handover has three phases: initiation, decision and execution. The handover can be initiated and executed for various reasons and can be divided into different kinds according to handover control, connection establishment and connection transference schemes.

The inter-segment handover strategy is critical in the design of IP-based terrestrial /satellite mobile communications networks. It not only enables the mobile terminal to change the radio link that connects the mobile terminal and the network to maintain a good QoS while the mobile terminal moves into a different network, but also allows the mobile user to obtain a new IP address and inform the home agent and corresponding user this change to receive the packets via the new link.

In order to design a fast and efficient inter-segment handover protocol, a soft, mobile-controlled handover has been selected as the handover method according to the characteristics of the inter-segment handover. The reasons are as follows:

1. The costs of the satellite link and terrestrial link are different and the user should have the right to choose the suitable segment;
2. The user can select a suitable segment according to the different geographical environments, which reduces unnecessary handover.
3. The signalling exchanges between the MT and network can be reduced to a minimum. The mobile-controlled handover will reduce the signalling load and the signalling delay and improve the efficiency of the network.
4. In the soft handover, the new link is established before the old link is released, and the data can be transmitted in both links simultaneously. To keep a good QoS, especially reducing the packet loss during the handover, the soft handover is also selected.

Inter-segment handover can occur in either of two directions: the satellite-to-terrestrial handover and the terrestrial-to-satellite handover. Since the procedures of these two kinds of inter-segment handover are very similar, only the satellite-to-terrestrial inter-segment handover protocol is described in this paper. The procedure is as follows:

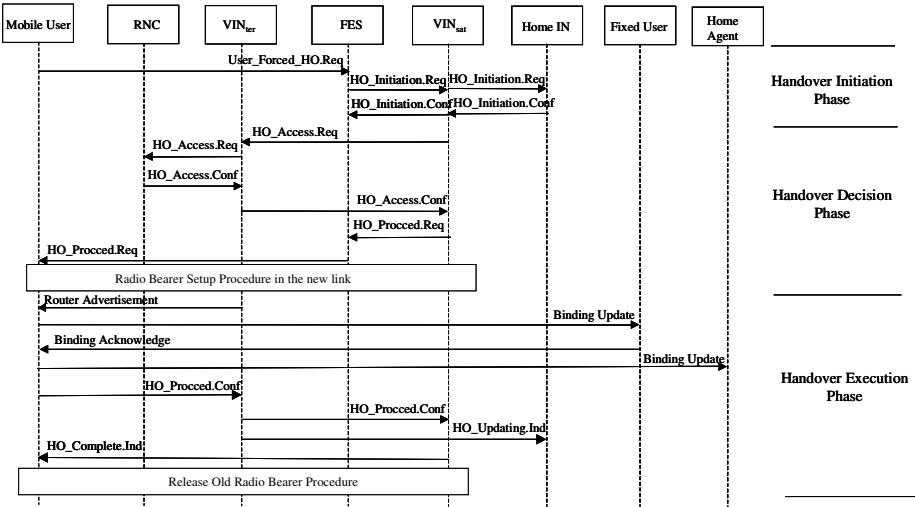


Fig. 3. Signalling Flow for the Satellite-to-Terrestrial Inter-Segment Handover Procedure

During the handover initiation phase, when the mobile terminal detects a QoS change, it can access a terrestrial link and send a handover initiation request message to the network. When the FES receives this request, it forwards the message to the visited intelligent network and the latter relays the message to the home intelligent network. The home intelligent network will check the request and identifies the target terrestrial cell, and sends relative information to the visited intelligent network in the satellite access network.

Then the visited intelligent network in the satellite access network sends a resource access request to its terrestrial counterpart. If the requested resource is available, the RNC in the terrestrial network is asked to reserve the radio resource. The intelligent network in the satellite access network sends a handover command, which includes the information about the new radio bearer, to the mobile terminal via the old signalling link.

When the mobile terminal receives this message, it enters the handover execution phase. Firstly, it initiates the procedure of the new radio bearer setup, and a setup message is sent to the RNC via a BTS. Then the mobile terminal starts to establish a radio bearer in the new link to transport signalling and user data packets. After establishing the terrestrial radio bearer, the mobile terminal is connected to both the old and new bearers at the same time. By using mobile IP, the mobile user can obtain a new care-of-address, and then send messages to the home agent and the corresponding user to update their binding caches according to the optimal routing principle. Thereby, the new packets sent from the corresponding user can be delivered to the mobile user via the new link. After achieving this, the mobile terminal switches the connection from the old satellite link to the assigned terrestrial bearer. Finally, the mobile terminal releases the old satellite radio bearer.

5 Performance Analysis

The performance of the proposed signalling protocols are evaluated in the terms of the QoS parameters such as: protocol execution delay, throughput, and handover failure probability.

Delay is always a very important QoS parameter to evaluate the performance of signalling protocols. We will use the protocol execution delay to test the proposed inter-segment handover protocol. The protocol execution delay can be calculated as follows:

Let

T_1 = overall transmission delay

T_2 = overall propagation delay

T_3 = overall waiting delay

T_4 = overall processing delay

If T = protocol execution delay,

$$T = \sum_{i=1}^4 T_i. \quad (1)$$

The mean transmission delay is the average time for a message being transmitted to be pushed onto the channel. It depends on message lengths and the bit rates on the transmission links. The overall transmission delay is the sum of the transmission delays for all the messages being transmitted among different nodes.

The mean propagation delay is the average time for messages take to be propagated, and it depends on the distance between nodes. The overall propagation delay is the sum of the propagation delays for all the messages being transmitted among different nodes.

The mean waiting delay is the average time a message waits (is queued) in the system. It depends on the traffic load, the number of users, the number of servers, and the sizes of buffers. The traffic is generated by two traffic generators: one is a signalling traffic generator, and the other is background traffic generator, both of which have an exponential inter-arrival time distribution. We have also assumed that the service time of packets follows an exponential distribution in the analytical model. This will be approximately true since packet length has been taken to be exponentially distributed for the background traffic, and the signalling traffic and background traffic have the same mean values, although the former has a generally distributed packet length. The analytical model used is based on an $M/M/1$ queue with infinite buffer and therefore the waiting time of a single message in the queue can be calculated according to the following formula [5]:

$$T_{\text{waitting}} = \frac{1}{\mu - \sum_i^N \lambda_i}. \quad (2)$$

where

- λ_i The mean arrival rate from the i^{th} source
- μ Mean service rate

The processing delay is the time for a message being processed in a node, and includes the delays resulted from the encapsulation, decapsulation, routing and all the operations related to the processed message.

Since throughput is another important QoS parameter, it is also used to judge the performance of the proposed inter-segment handover protocol.

The handover failure probability is used to test the inter-segment handover protocol under different traffic and pre-set handover limitations. To keep a satisfied QoS, a handover should be performed as fast as possible, which means the signalling messages exchanged between the mobile user and the network should be minimised. To evaluate the handover failure probability of the inter-segment handover, a time limit is pre-set and the delay is measured from the mobile user sending a first message to the network to initiate an inter-segment handover till the “HO_complete.Ind” message being received. If the delay beyond the pre-set handover limit, it can be regarded as a failed handover.

The commercial software package OPNet is used to simulate the behaviour of the inter-segment handover protocol in an IP-based integrated terrestrial/satellite environment.

The numerical assumptions made are given in the Table 1 and the satellite parameters have been chosen to be consistent with a Medium Earth Orbit (MEO) link.

Table 1 Parameters Used in the Simulation

Parameter	Value
Mean Service rate	100000bit/second
Radio link bit rate	2000000bit/second
Signalling bit rate for transmitters and receivers	200000bit/second
Processing delay for each message	0.5ms
Propagation delay between MT and FES	70ms
Propagation delay between MT and RNC	1ms
Propagation delay between VIN_{sat} and HIN	1ms
Propagation delay between VIN_{ter} and HIN	1ms
Propagation delay between VIN_{sat} and VIN_{ter}	1ms

The results show that the protocol execution delay, throughput, and handover failure probability are highly dependent on the traffic loads.

Figure 4 shows the protocol execution delay of the inter-segment handover. Because the propagation delay of the satellite link is much longer than that of terrestrial link, the number of messages exchanged via the satellite link dominates the overall propagation delay during handover and also has a great impact on the waiting delay. Furthermore, it influences the protocol execution delay. If the users perform a satellite-to-terrestrial handover, when the traffic load changes from 5 handover/second to 20 handover/second, the delay increases from 344ms to 457ms. However, when the traffic load changes from 20 handover/second to 23 handover/second, the delay increases from 457ms to 704ms. The results show that the satellite-to-terrestrial handover is much faster than terrestrial-to-satellite handover. For example, when the traffic load is 20 handover/second, the satellite-to-terrestrial handover is 457ms and terrestrial-to-satellite handover is 1140ms. It conforms with

the common opinion that terrestrial communications is always the first choice, and satellite communications serves as an alternative in certain situation, such as when the terrestrial network is not available. If possible, the user can quickly perform a satellite-to-terrestrial handover when in pursuit of a higher QoS and lower cost. Figure 4 also shows that the simulation results match reasonably well the analytical results predicted by a model incorporating equations (1) and (2), thus giving a degree of confidence in the simulation.

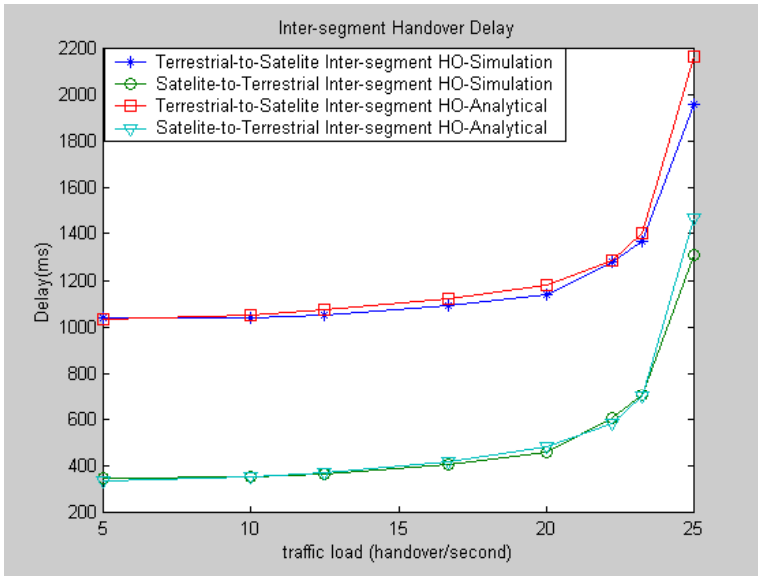


Fig. 4. Effects of Load on the Inter-segment Handover Protocol Execution Delay

Together, Figures 4 and 5 shows that the growth of traffic load leads to an increase in throughput, but at the cost of an increased handover delay. Since an infinite buffer is used in the queue, there is no packet loss. As the traffic load increases, the throughput increases nearly linearly.

Figure 6 shows the effects of the network traffic and pre-set handover limit on the handover failure probability. Because the handover execution delay in the terrestrial-to-satellite inter-segment handover is much longer than that of the satellite-to-terrestrial inter-segment handover, a higher pre-set limit is used in the former. The handover time limits 400ms, 450ms and 500ms have been used for satellite-to-terrestrial inter-segment handover, and handover limits 1100ms, 1150ms and 1200ms for terrestrial-to-satellite inter-segment handover. Note that if satellite-to-terrestrial inter-segment handover is performed when the traffic load is 12.5 handover/second, the handover delivery failure probability is 5.6%, 2.4%, and 1.1% for handover limit 400ms, 450ms and 500ms respectively. The heavier the traffic load, the higher handover delivery failure probability, and the higher the pre-set limit, the lower the handover delivery failure probability, so a compromise is required.

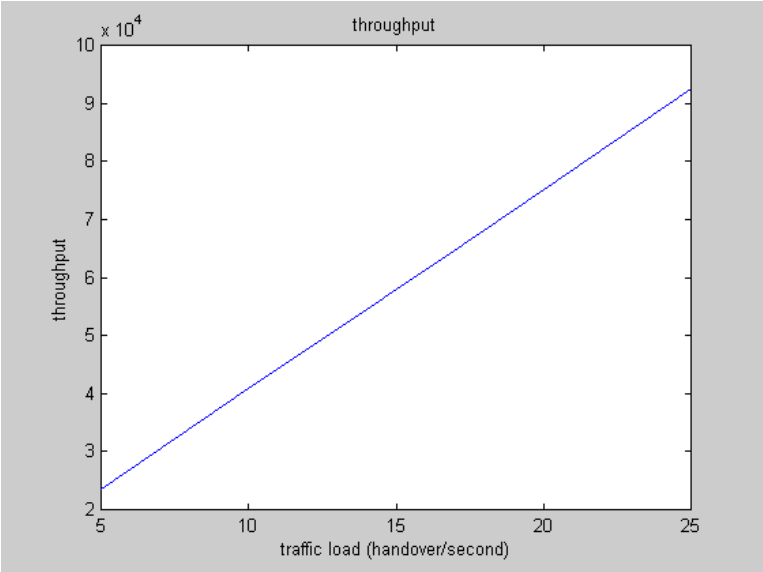


Fig. 5. Effects of Load on Throughput

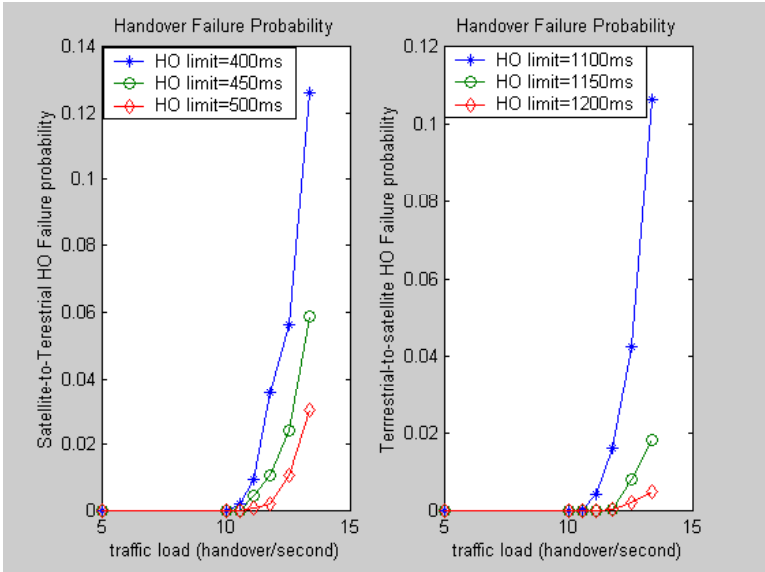


Fig. 6. Effects of Load on the Handover Failure Probability

6 Conclusion

In this paper, a new architecture for an IP-based integrated terrestrial/satellite mobile communications network is proposed, an inter-segment handover protocol is described. The performance of the proposed inter-segment handover protocol is also analysed and evaluated. The results show that both the protocol execution delay and throughput are highly dependent on the traffic load. The handover failure probability depends on a trade-off of the network traffic and pre-set handover limit. In this content, it is shown that a much higher limit has to be used in the terrestrial-to-satellite handover than for the satellite-to-terrestrial handover. This is due to the longer propagation delay on the satellite link and the larger number of messages that be exchanged via satellite link in the terrestrial-to-satellite inter-segment handover protocol. The proposed architecture and inter-segment handover protocol will provide a potential solution to seamlessly integrate the terrestrial and satellite networks in the future IP-based global mobile communications networks.

References

1. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects General UMTS Architecture (3G TS 23.101 Version 3.0.1) (1999-04).
2. ITU-T Recommendation Q.1701: "Framework for IMT-2000 Networks", March 1999.
3. C. Perkins, IETF RFC 2002, "IP Mobility Support", 1996.
4. ITU-T Recommendations Q.1711: "Network Functional Model for IMT-2000", March 1999.
5. M.E. Woodward: "Communication and Computer Networks – Modelling with discrete-time queues", Pentech Press, 1993.

Performance Evaluation of Voice-Data Integration for Wireless Data Networking*

Mainak Chatterjee¹, Sajal K. Das¹, and Giridhar D. Mandyam²

¹ Center for Research in Wireless Mobility and Networking (CReWMaN)
Department of Computer Science and Engineering

The University of Texas at Arlington

Arlington, TX 71019, USA

{chat,das}@cse.uta.edu

² Nokia Research Center

6000 Connection Drive

Irving, TX 75039, USA

giridhar.mandyam@nokia.com

Abstract. In this paper, we study the effect of integrating voice and data services in wireless data networks. In particular, we evaluate the performance on cdma2000 system which is currently one of the strongest contenders for wireless data networking technology. Since the true nature of the wireless data is still not known, we use a mix of Poisson-distributed voice packets and Pareto-distributed data packets. The ratio of the mix is varied and the performance is studied with respect to channel utilization, waiting time, and blocking probability. The effect of the shape parameter of the Pareto-distributed data packets on the system performance is also studied.

Keywords: wireless voice/data integration, Pareto distribution, cdma2000 networks, resource allocation.

1 Introduction

All the emerging wireless data networking technologies today rely on the Internet Protocol (IP) because IP is still the most dominant internetworking protocol. Moreover, the already existing Internet infrastructure should be exploited as much as possible to defray the cost of overlaying wireless technology. The advances in IP coupled with the provisioning for quality of service (QoS) for multimedia applications makes IP a good choice for cellular providers to deliver the service to the already existing huge customer base.

The integration of voice services from the cellular domain and data services from the IP domain, is still a major challenge because of the differences in the characteristics and requirements of both the services. Voice services are delay-sensitive and aim to provide equal service to all users regardless of their location

* Supported in part by Texas Advanced Research Program grant (TARP-003594-013), Texas Telecommunication Engineering Consortium (TxTEC), and Nokia Research.

in the cell within the cellular architecture. Moreover, voice calls last for a few minutes with silence between talk spurts most of the time. A better channel utilization can thus be obtained by sharing the channel resources with other voice terminals. These features result in allocation of power depending on the individual user's voice activity. Voice services until now have been optimized for only voice telephony which requires continuous bit-stream type service with no delays. Additionally, a relatively modest data rate is sufficient for high-quality voice service, and voice users cannot substantially benefit from higher data rates.

In contrast, data services work well with discontinuous packetized transmission and can tolerate more delay. Packet data systems are aimed at maximizing the throughput. Given various data rate requirements for different data users, the goal is no longer to serve everyone with equal power and equal grade of service. Rather, the goal is to allocate users the maximum data rate that each can accept based on application needs and wireless channel conditions.

To provide multimedia services through wireless channels, it is also important that certain quality of service (QoS) parameters as specified by the applications are satisfied such that a balance between the number of users served and their degree of satisfaction is obtained [3]. Certain schemes like efficient admission control, optimal resource management and good error control are adopted to achieve this. The requirements for future multimedia services include higher capacities, increased spectral efficiency, higher speeds and differentiated services.

cdma2000, which is an evolution from IS-95 standard, is currently one of the strongest contenders for wireless data networking [6,11]. It provides next-generation capacity while maintaining backward compatibility [12]. In this paper, we investigate the performance of cdma2000 to support both voice and data services. Five classes of services are considered depending on their bit rate requirements. For modeling voice services, we use Poisson distribution and hence the inter-arrival time for these packets is exponentially distributed. For data services, the session duration as well as the inter-arrival time for packets are modeled using Pareto distribution [10]. Different shape parameters are considered to account for the variability in the burstiness of the data packets. The ratio of the load due to voice services and data services is varied for the entire possible range, for example, a mix from 0% voice and 100% data to 100% voice and 0% data. The metrics we considered for the evaluation are channel utilization, waiting time and blocking probability.

The rest of the paper is organized as follows. In Section 2, we discuss the nature of wireless data. Section 3 presents our approach for modeling both voice and data services. Section 4 describes our simulation model while Section 5 discusses the experimental results. Conclusions are drawn in the last section.

2 Nature of Wireless Data

Modeling network traffic using a Poisson or Markovian arrival process is common because of its theoretical simplicity. It also has some favorable properties like the smoothening of the total traffic by statistical aggregation of multiple

Markovian arrival processes, which are individually bursty in nature. Careful statistical analysis of data collected from experiments on Ethernet LAN traffic [8] for long durations has shown that such data actually exhibit properties of *self-similarity* [1,10] and that there is long-range dependencies among the data. It is observed that such a traffic is bursty over a wide range of time scales which can usually be generated by heavy-tailed distributions with infinite variance [4]. Pareto distribution is such a distribution with heavy tail and large burstiness. This self-similar nature of Ethernet traffic is different both from the conventional traffic models and also from the currently considered formal methods of packet traffic [8]. However, there is still considerable debate [5] over the actual modeling of network traffic because it has serious implications on the design and analysis of networks. Analysis using self-similar models generally ignores the time-scale in which the experiments are performed. The finite range of the time periods of our observations makes it necessary to study and model network traffic as not strictly self-similar. The amount of correlation that we should consider should not only depend on the correlation nature of the source traffic but also on the time scale which is specific to the system under consideration.

Just considering either of the two distributions (Poisson and Pareto) will not truly represent the nature of wireless data and therefore characterization becomes difficult. In fact, till date there exists no unified model which represents the true nature of wireless data. Services like file transfer, e-mail and store-and-forward facsimile – usually known as short-messages services (SMS) – are relatively short and can be represented by a Poisson model [13]. Interactive data services can be modeled as a queue of packets at each source with a random arrival process into the queue. The expected session length of these services is 1-2 minutes, which is really a short interval to see the effect of any long range dependencies. It might even be difficult to find any kind of correlation among the data pattern within that interval of time.

3 Our Voice/Data Model

It is our belief that wireless traffic would not strictly follow Poisson or Pareto distributions, but will have components from both. Now the question arises about the percentage contribution of voice and data. It has been recently observed that there is a steady increase in the number of data users. What might be appropriate is to evaluate a system where all possible combinations of voice and data components for a given load are considered. Let us now consider the two models.

3.1 Voice Model

We assume that active users produce and transmit voice packets at a certain rate and inactive users do not transmit at all. A voice call shows periods of activity and inactivity. We model the duration of both *talk spurts* (activity) and *gaps* (inactivity) as exponentially distributed. If the mean duration for the talk

spurts is τ_t and the mean duration for the gaps is τ_g , then *activity*, A , is defined as

$$\text{activity} = A = \frac{\tau_t}{\tau_t + \tau_g}.$$

If the length of the talk spurts are given by the random variable $\mathcal{T}$, then

$$\mathcal{T} = -\tau_t \times \ln(1 - U)$$

where U is uniformly distributed between 0 and 1. We also assume that the duration of a voice session is exponentially distributed with a certain mean.

3.2 Data Model

Current Internet traffic has been shown to be self-similar, which means that such traffic is bursty over a wide range of time scales. As Ethernet traffic was shown to be different from conventional traffic models, we use a heavy-tailed Pareto distribution for modeling data traffic. The active data spurts are assumed to be independent and identically distributed according to the Pareto distribution with shape parameter α and scale parameter k . The cumulative distribution function $F(t)$ for Pareto distribution is given by

$$F(t) = 1 - \left(\frac{k}{t}\right)^\alpha.$$

The burstiness of the data packets can be controlled by the changing shape parameter α . It is observed in [4] that for all practical purposes $1 \leq \alpha \leq 2$. If the length of the data spurts are given by the random variable $\mathcal{D}$, then $\mathcal{D}$ can be obtained as

$$\mathcal{D} = k \times e^{-\frac{\ln(U)}{\alpha}}$$

It is also assumed that the duration of a data session is Pareto distributed with a certain mean.

3.3 Multiplexing Voice and Data

As mentioned earlier, wireless data will have packets from the IP domain as well as the telephony domain. Therefore, the base station would receive both voice and data multiplexed. For any resource allocation scheme, the base station has to consider the multiplexed stream. If we define *voice fraction* K as

$$K = \frac{\text{voice load}}{\text{voice load} + \text{data load}}$$

then $(1 - K)$ is the data fraction of the load. The two extreme cases are $K = 0$ implying no voice component and $K = 1$ implying no data component. Note that at any point of time, the total load in the system remains the same.

3.4 Performance Metric

We do not consider any prioritization between voice and data users in our service admission scheme. This may not be true in practice. Different operators may assign priority between voice and data users based on the potential revenue provided by either type of subscriber. For admitting a service into the system, we compare the current load to the total channel capacity C fixed at 100. A service is blocked or denied admission into the system if the capacity with the inclusion of the new service exceeds C . The percentage of the blocked services gives the *blocking probability*. Once a service is admitted, it generates bursts depending on the nature of the service. For every burst, the scheduler at the base station tries to allocate the required number of traffic channels. Every burst has to contend for the required number of traffic channels before it could go into the transmission phase. The burst is not allocated any traffic channel unless and until the required number of traffic channels are available. This contention process incurs some delay. We define *waiting time* (which is a measure of the delay) as the average of the delays incurred by the bursts. A service which acquires some traffic channels for the transmission of a burst, relinquishes those channels soon after the successful transmission of the burst. We also define *channel utilization* as the fraction of the time the traffic channels actually transmit packets.

4 Simulation Model

For the purpose of studying the effectiveness of simultaneously supporting voice and data traffic, we choose to use the cdma2000 system [15]. cdma2000 includes sophisticated medium access control (MAC) features which can concurrently support multiple data and voice services, thus effectively supporting very high data rates (upto 2 Mbps). cdma2000 extends support for multiple simultaneous services much more than the services provided by IS-95-B. It does so by providing much higher data rates and a sophisticated multimedia QoS capability to support multiple voice/packet, circuit/data connections with differing QoS requirements. The design of cdma2000 allows for deployment of the 3G enhancements while maintaining the current 2G support for IS-95 in the spectrum that an operator has today. It is also compatible with the IMT-2000 spectrum bands [2], so operators acquiring new spectrum will be able to experience the benefits of cdma2000 as well.

In our simulation experiments, we considered 5 classes of traffic depending on their data rates. The data rates considered are in the ratio of 1:3:6:9:12. This is because, cdma2000 has a multi-carrier feature in which a service can be allocated more than 1 (3, 6, 9 or 12) traffic channels for data transmission. Requests are made for both types of service connection establishment at a certain arrival rate λ . Request for connection establishment by all the classes are assumed to be equally probable. If a service is admitted into the system, then the service specifies its requirement profile in terms of its type (voice/data) and the bit rate requirement. The generation of bursts (a collection of packets) for a voice service is Poisson distributed and that of data service is Pareto distributed.

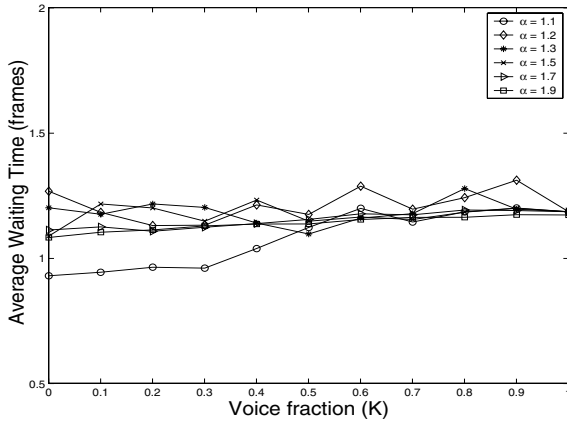


Fig. 1. Average Waiting Time

The session lengths for voice and data services are also Poisson and Pareto distributed, respectively. We restrict the duration of a data service to 10 times the average duration. The other parameters used for simulation are shown in Table 1.

Table 1. Parameters used for simulation

No. of Traffic Channels (C)	100
Service Arrival Rate (λ)	0.005/frame
Activity (A)	0.5
Mean Voice Session	2 mins
Mean Data Session	2 mins
Maximum Data Session	20 mins
Frame Duration	20 ms
Mean Burst Length	30 frames

Static allocation of code channels to a small number of users leads to an inefficient use of the CDMA air interface capacity. Dynamic allocation of the code channels to the arriving bursts makes it possible to share the channels among large number of users without hampering their QoS requirements. The allocation of the codes is done at the beginning of each frame. This is in contradiction to the present cdma2000 system which cannot continuously reallocate codes at the beginning of each frame. However, the enhancements proposed to IS-2000 called 1XTREME [9] has this feature implemented and we assume that our current system can dynamically reallocate codes at the beginning of each frame. The number of traffic channels that can carry data simultaneously is limited by the

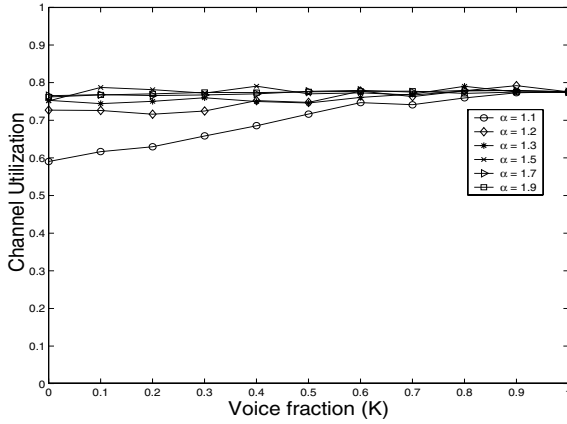


Fig. 2. Channel Utilization

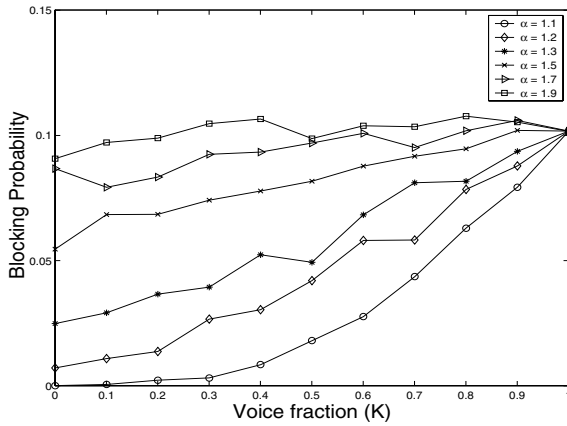
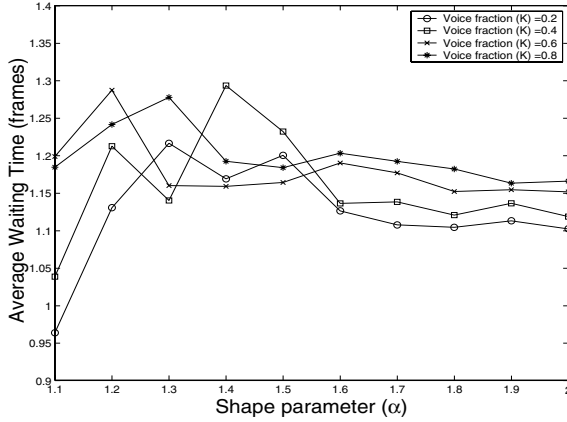
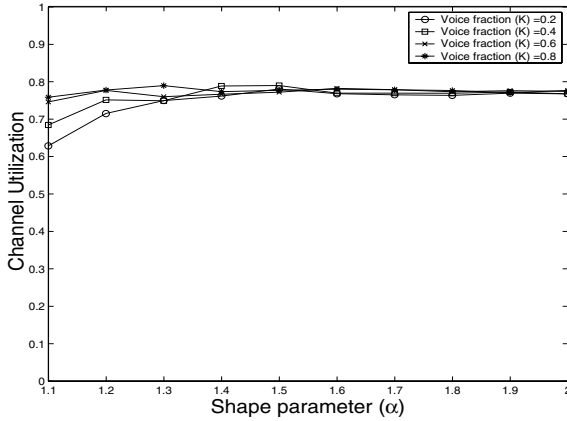


Fig. 3. Blocking Probability

number of traffic channels available and not due to noise or multiple access interference (MAI) [7]. This can be justified by the fact that the interference amongst the users are proportional to the additive signal strength, so the noise interference can be approximated as a linear function of the number of traffic channels.

5 Experimental Results

Figure 1 shows how the waiting time varies as the *voice fraction* (K) changes from a minimum of 0 to a maximum of 1. It is seen that for lesser values of

**Fig. 4.** Average Waiting Time**Fig. 5.** Channel Utilization

the shape parameter α , the behavior is erratic, i.e., the average waiting time is non-monotonic with the voice fraction. The channel utilization (Figure 2) almost remains the same except for $\alpha = 1.1$. The blocking probability (Figure 3) increases with the increase in voice component. It can be noted that at $K = 1$, all plots corresponding to different α , converge. This is because, there is no data component and the voice component offers the same load.

We also study the effect of the shape parameter α on the waiting time, channel utilization and blocking probability. The range considered is $1 \leq \alpha \leq 2$. The voice fraction K was maintained at 0.2, 0.4, 0.6 and 0.8. A non-monotonic nature in the waiting time is observed from Figure 4 for $\alpha \leq 1.6$. The channel utilization (Figure 5) is almost the same for the different values of K . The

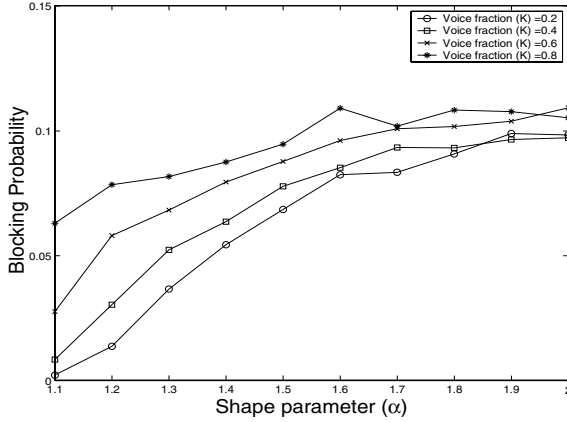


Fig. 6. Blocking Probability

blocking probability increases monotonically with the increase in α . If we have to provide strict delay guarantees, then we will have to admit fewer number of services into the system, thereby blocking most of them. But the channel utilization in that case will decrease. We can find the desirable operating point so that the system is able to perform within certain bounds. From these results we can decide on the fraction of voice and data which would give the desired level of performance. In other words, we can determine if a service is to be admitted into the system if all the QoS requirements of all the on-going services are to be satisfied.

6 Conclusion

As cdma2000 grows to be one of the most important data networking technologies, it is important to see how effective it will be in terms of supporting integrated voice and data services. Since the true nature of the wireless data is still not known, we used a mix of Poisson-distributed voice packets and Pareto-distributed data packets. We conducted simulation experiments and generated services of different types with different requirements. The proportion of the load due to voice and data services were varied for the entire spectrum and the performance of the system with respect to channel utilization, waiting time and blocking probability were studied. The effect of the shape parameter for the Pareto-distributed data packets were also studied. These results give insight into the performance as would be expected from a system dealing with both voice and data. Also, for a system to perform with certain QoS guarantees, the operating point for the system can be determined.

References

1. M.E. Corvella and A. Bestavros, *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*, IEEE Transaction on Networking, Vol 5, No. 6, Dec 1997, pp 835-846.
2. E. Dahlman, B. Gudmundson, M. Nilsson, and A. Skold, *UMTS/IMT-2000 based on wideband CDMA*, IEEE Communications Magazine, Vol. 36 Issue 9, Sept 1998, pp 70-80.
3. J.S. Evans and D. Everitt, *Effective bandwidth-based admission control for multi-service CDMA cellular networks*, IEEE Transaction Vehicular Technology, vol 48, Jan 1999, pp 36-46.
4. Z. Hadzi-Velkov and L. Gavrilovska, *Performance of the IEEE 802.11 Wireless LANs under Influence of Hidden Terminals and Pareto Distributed Packet Traffic*, IEEE International Conference on Personal Wireless Communications, 1999, pp 221-225.
5. M. Grossglauser and J.C. Bolot, *On the Relevance of Long-Range Dependence in Network Traffic*, IEEE Transaction on Networking, Vol 7, No. 5, Oct 1999, pp 629-640.
6. D.N. Knisely, S. Kumar, S. Laha, and S. Nanda, *Evolution of wireless data services: IS-95 to cdma2000*, IEEE Communications Magazine, Vol. 36 Issue 10, Oct 1998, pp 140-149.
7. W.C.Y. Lee, *Overview of cellular CDMA*, IEEE Transaction Vehicular Technology, vol 40, May 1991, pp 291-302.
8. W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, *On the Self-Similar Nature of Ethernet Traffic (Extended Version)*, IEEE Transaction on Networking, Vol 2, No. 1, Feb 1994, pp 1-15.
9. G.D. Mandyam and G. Fry, *1XTREME: A Step Beyond 3G*, IEEE DSP Workshop, Oct 2000.
10. V. Paxson and S. Floyd, *Wide Area Traffic: The Failure of Poisson Modeling*, IEEE Transaction on Networking, Vol 3, No. 3, Jun 1995, pp 226-244.
11. R. Prasad and T. Ojanpera, *A survey on CDMA: evolution towards wideband CDMA*, Proceedings of IEEE 5th International Symposium on Spread Spectrum Techniques and Applications, Vol. 1, 1998, pp 323-331.
12. Y.S. Rao and A. Kripalani, *cdma2000 mobile radio access for IMT 2000*, IEEE International Conference on Personal Wireless Communication, 1999, pp 6-15.
13. A. Sampath and J.M. Holtzman, *Access control of data in integrated voice/data CDMA systems: Benefits and Tradeoffs*, IEEE Journal on Selected Areas in Communication, Vol. 15, No. 8, Oct 1997, pp 1511-1526.
14. TIA/EIA/IS-2000.3-A, *Medium Access Control (MAC) Standard for cdma2000 Standards for Spread Spectrum Systems*.
15. TIA/EIA/IS-2000.3-A, *Medium Access Control (MAC) Standard for cdma2000 Standards for Spread Spectrum Systems*.

A Hard Handover Control Scheme Supporting IP Host Mobility

Yoshihito Takahashi, Noriteru Shinagawa, and Takehiko Kobayashi

YRP Mobile Telecommunications Key Technology Research Laboratories Co., Ltd.
3-4 Hikari-no-oka, Yokosuka, Kanagawa, 239-0847, Japan
{ytaka, shina, koba}@yrp-ktrl.co.jp

Abstract. This paper proposes a hard handover control scheme that supports Internet Protocol (IP) host mobility by means of IP address translation servers that translate an IP address into another, and a handover server that registers locations of IP mobile hosts and remotely controls the servers. A functional comparison of the proposed scheme and conventional ones such as Mobile IPv4, Mobile IPv6, and Cellular IP is also presented in this paper.

1 Introduction

The growing use of portable computers is increasing the demand for using the Internet in mobile situations. Over the Internet, packets are transferred from source Internet Protocol (IP) hosts to destination IP hosts. Each packet has a source IP address and a destination IP address. However, because the IP addresses have the dual role of serving as an identifier that specifies the IP host and serving as an identifier that specifies the location of the IP host, the IP address must be changed as the IP host moves from place to place. Since communication generally cannot continue if the IP address changes, use of the Internet in mobile situations has not been possible. Mobile IPv4[1-4], Mobile IPv6[5], and Cellular IP[6] have been proposed to solve this problem. The Mobile IPv4 scheme has the shortcomings described in (1) through (3) below:

- (1) There is a problem of triangular routing resulting in path redundancy.
- (2) When the Home Agent (HA) to which a mobile IP host is usually connected is located at a distance, updating of the host's registration takes time. For example, if a mobile IP host whose HA is in France is taken to Japan, it could take a long time to update the registration, and timeout or loss of user packets could frequently occur.
- (3) Because inner IP packets are encapsulated in outer IP packets, the outer IP header size is increased by the size of inner IP header.

The Mobile IPv6 scheme also has the drawbacks described by (4) through (6) below:

- (4) For the initial packet transfer, it also has a path-redundant triangular routing problem.
- (5) It also has the long-distance registration problem.

- (6) A home address, which is an IP address assigned in the HA to a mobile IP host, and other information are placed in extended header, which increases the size of the header.

The Cellular IP scheme also has the shortcomings described in (7) through (9) below:

- (7) The routing entity has to be modified.
- (8) Cases in which it is necessary to change paths in a part of the network that comprises switching entities cannot be handled.
- (9) Dummy packets must be sent in order to maintain the routing cache mapping, even when there are no user data packets to be sent, thus wasting bandwidth in both the wired and wireless links.

In order to solve these problems of the Mobile IPv4, Mobile IPv6 and Cellular IP schemes, this paper proposes a hard handover control scheme that supports IP host mobility. In this scheme, IP address translation servers controlled by a handover server are used to translate one IP address into another, which enables rerouting due to handover.

2 Proposed IP Packet Transfer Control Scheme

2.1 Network Configuration

An example configuration of the mobile network that we propose for supporting IP host mobility is illustrated in Fig. 1. A mobile terminal (MT), which is an IP host moving in cellular service areas, communicates with a communication server, which we refer to as the Correspondent Entity (CE), via a wired backbone network. The wired network consists of IP nodes (IPN1 through IPN4), an IP address translation server (ATS4), base stations 1, 2 and 3, which are equipped with IP nodes (IPN5 through IPN7) and IP address translation servers (ATS5 through ATS7), and a handover server (HS). The ATSs are incorporated in each mobile-network-edge-entity connected to IP terminals, IP servers, or other IP networks. Every ATS has an IP address translation table designated by the HS. Based on the table, the ATS rewrites the destination and source addresses of IP packets it receives and forwards the packets to their new destinations. The HS manages the locations of the MTs based on the location information sent from them and remotely controls the ATSs by using regularly updated network status information. In the proposed scheme, an IPN can be any node that is capable of transferring IP packets, including conventional IP routers, IP switches, and ATM switches supporting IP-over-ATM protocols. The MTs and CEs require no special or additional IP transfer functions except those already in the IP modules. Furthermore, modification of the routing entities (*i.e.*, rewriting of the routing table at the time of handover of the MTs) is not needed, as it is with the Cellular IP scheme.

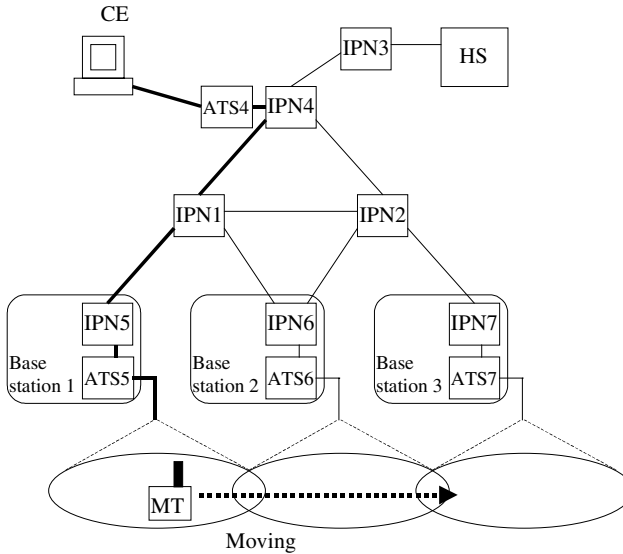


Fig. 1. Network configuration of a mobile network supporting IP host mobility.

2.2 IP Packet Flow

The downlink IP packet flow before a handover is shown in Fig. 2. The MT is located in the radio zone of base station 1, and IP packets sent from the CE to the MT are transferred via base station 1. The IP packets have IP address information “Dst = M, Src = C” in their headers. When they are received at ATS4, the information is replaced with “Dst = X5.a, Src = X4.b” and the packets are forwarded to ATS5. When they are received at ATS5, the information is replaced with “Dst = M, Src = C” and the packets are forwarded to the MT. In this way, the IP packets are transferred from the CE to the MT via base station 1.

The downlink IP packet flow after the handover is illustrated in Fig. 3. The MT is now located in the wireless zone of base station 2. Thus, IP packets sent from the CE to the MT have to be transferred via base station 2. The HS sends out commands to the ATSSs and remotely controls the IP address translation so as to make such rerouting possible. When the packets having IP address information “Dst = M, Src = C” in their headers are received at ATS4, the information is replaced with “Dst = X6.a, Src = X4.b” and the packets are transmitted to ATS6. When the packets are received at ATS6, the information is rewritten into “Dst = M, Src = C” and the packets are forwarded to the MT. In this way, the IP packets are transferred from the CE to the MT via base station 2.

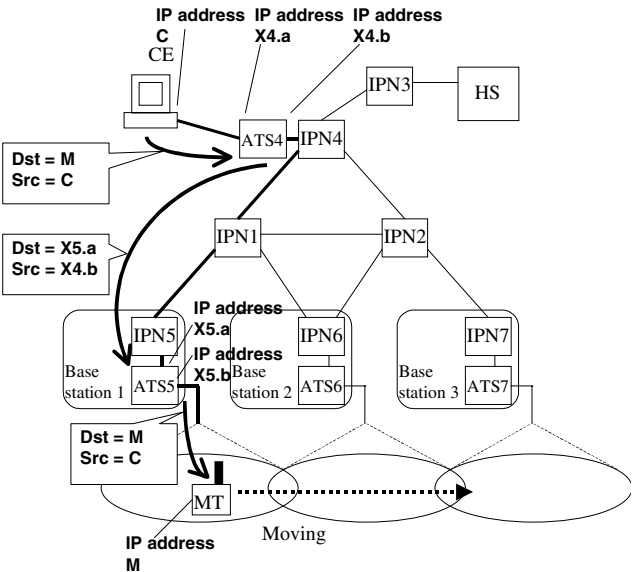


Fig. 2. Downlink IP packet flow before handover.

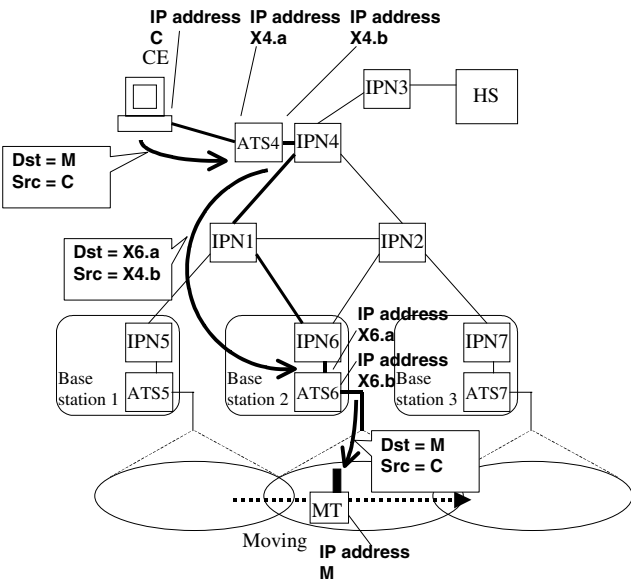


Fig. 3. Downlink IP packet flow after handover.

Since the proposed scheme does not involve a HA, the triangular routing problem that arises with the Mobile IPv4 and IPv6 schemes does not occur. Furthermore, since the proposed scheme involves only rewriting of the IP address in the IP header, there is no increase in header size as there is with the Mobile IPv4 and IPv6 schemes. Further still, there is no rewriting of the routing tables of the routing entities as there is with the Cellular IP scheme, thus the rerouting that accompanies handover is easy, even when the IPN is an IP switch.

2.3 Control Performed by the Handover Server

The HS registers the current location of the MTs and remotely controls the IP address translation of the ATSS. The location registration of an MT at the HS is illustrated in Fig. 4. When base station 2 detects an MT in the layer below the IP layer (*e.g.*, wireless control), ATS6 sends a control packet to the HS to register the MT's location. The HS then updates the registered location of the MT based on the information in the control packet.

Figure 5 shows how IP address translation update commands are sent to the ATSS from the HS. After receiving the packets to register MT's location from ATS6, the HS computes new appropriate transit entities between the CE and MT (via base station 2), based on the network status information it maintains, and sends out the transit information in the form of a list of ATSS. The HS also creates an IP address translation update command that specifies how the IP address translation should be performed at each ATS that is now the transit entity, and then it sends the command directly to the ATSS. When the transit entities receive those packets, they update their own IP address translation table. In the example shown in Fig. 5, ATS4 has possessed a translation table between $M \leftrightarrow X5.a$. Based on an update command from the HS to ATS4, it updates its table so as to translate between $M \leftrightarrow X6.a$. ATS5 also has possessed a translation table between $X5.a \leftrightarrow M$ and between $X4.b \leftrightarrow C$. According to an update command from the HS to ATS5, it deletes the contents of the translation table. ATS6 initially has possessed no entries in a translation table. Based on an update command from the HS to ATS6, it adds instructions to translate between $X6.a \leftrightarrow M$ and between $X4.b \leftrightarrow C$. By remotely controlling the ATSS in this way, the HS controls hard handover for the MTs.

In the proposed scheme, when a base station detects an MT in the layer below the IP layer (*e.g.*, wireless control), it registers the MT's location with the HS. The HS then initiates handover control, and performs the *hard-state* control, which is executed by issuing explicit control commands, of IP address translation. In the Cellular IP scheme, the mapping of the routing cache is automatically deleted if IP packets do not arrive within a certain period of time. This kind of control is referred to as *soft-state* control which is executed implicitly; *i.e.*, explicit control commands are not used. The *soft-state* control used in the Cellular IP scheme requires that dummy packets are sent continuously to maintain the old mapping of the routing cache. Meanwhile, since the proposed scheme does not use such *soft-state* control, no dummy packets have to be sent.

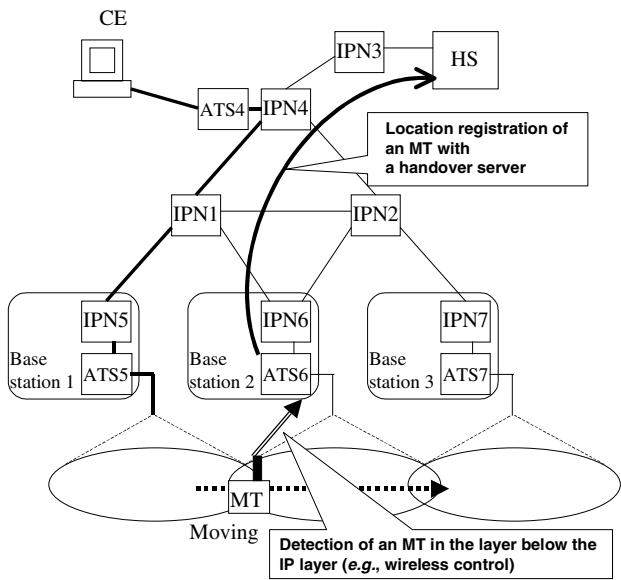


Fig. 4. Location Registration at the HS.

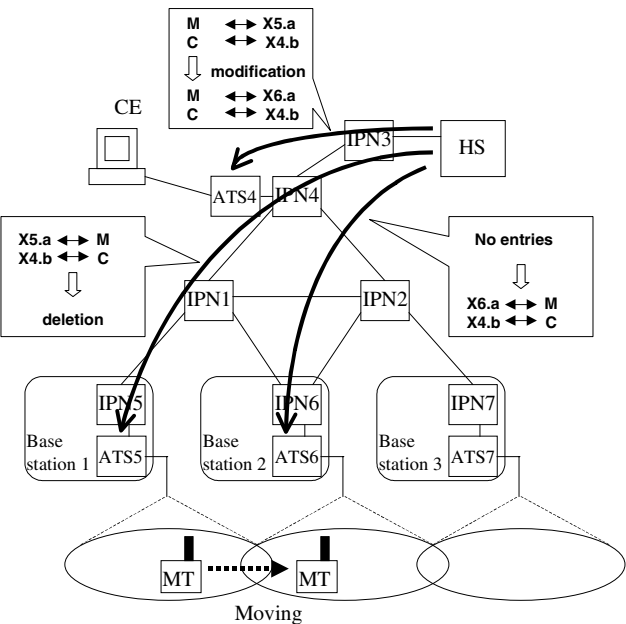


Fig. 5. Commands to update IP address translation tables.

3 Conclusion

This paper has proposed a hard handover control scheme that supports IP host mobility. This scheme introduces IP address translation servers and a handover server. The translation servers translate an IP address into another. The handover server registers the location of IP mobile hosts, and remotely controls the translation servers. This makes it possible to transfer IP packets to a new destination after handover. This paper also has shown how the proposed scheme solves the problems associated with the Mobile IPv4, Mobile IPv6 and Cellular IP schemes.

References

1. Perkins, C.: IP Mobility Support for IPv4. Draft-ietf-mobileip-rfc2002-bis-04.txt. (2001)
2. Perkins, C.: IP Encapsulation within IP. RFC2003. (1996)
3. Perkins, C.: Minimal encapsulation within IP. RFC2004. (1996)
4. Montenegro, G.: Reverse Tunneling for Mobile IP. RFC3024. (2001)
5. Johnson, D.B., Perkins, C.: Mobility Support in IPv6. Draft-ietf-mobileip-ipv6-13.txt. (2000)
6. Valko, A.G.: Cellular IP: A New Approach to Internet Host Mobility. Computer Communication review, Vol. 29, no. 1. (1999) 50-65

A Flexible User Authentication Scheme for Multi-server Internet Services¹

Woei-Jiunn Tsauro

Department of Information Management, Da-Yeh University, Changhua, Taiwan, R.O.C.
wjtsaur@mail.dyu.edu.tw

Abstract. Due to the rapid progress of information technology, computer systems with the client/server architecture have been becoming a new way in multi-user computing environments. For the environment of single server, the issue of remote login authentication has already been solved by a variety of schemes, but it has not been efficiently solved for multi-server Internet environments yet. In this paper, we will present an efficient smart card based remote login authentication scheme for multi-server Internet environments, which can verify a single password for logging multiple authorized servers without using any password verification table. The objective of the new scheme emphasizes that any client can get service grant from multiple servers without repetitive registration to each server. The proposed scheme's advantages include that not only repetitive registration for various servers is avoided, but also the network users can freely choose their preferred passwords and be deleted easily by the system. Moreover, security analyses about the impersonation and replay attacks on the proposed scheme validate the feasibility of the scheme.

Keywords: Communications security, Internet, Password authentication, Smart card, Lagrange interpolating polynomial.

1 Introduction

With the distributed nature of computer and network systems, the achievement of privacy and security has become increasingly important. In the past decade, various kinds of authentication mechanisms have been developed for protecting information or resources from unauthorized users [1-4]. Among them, password authentication is the most acceptable and widely used mechanism because of its inexpensive cost, ease of use, and simple implementation [3, 5, 6].

In a traditional password authentication scheme, each user is equipped with an identity number ID and a secret password PW . A valid ID and its corresponding password PW are required whenever a user wishes to enter the network system. The most straightforward authentication approach is to construct, in advance, a directory which stores each user's ID and the corresponding PW . Each network user, say U_i , submits his/her identity U_ID_i and password U_PW_i during the login phase when

¹ This work was supported by the National Science Council of Republic of China under contract NSC 89-2213-E-212-037.

attempting to enter the network system. The network system then searches through the password directory table to see if the submitted password agrees with that which is prestored in the table. If the result is yes, then U_i is recognized as an authorized user and is permitted to enter the network; otherwise, the login request is rejected. Apparently, from the viewpoint of memory protection, such an approach is too weak to ensure the security of the plain password table stored in the system. Also, it may introduce an additional burden on the system for managing the password table. Instead of directly storing the plain password table, alternative approaches [7, 8] can be applied as follows:

- (1) Encrypt users' passwords into test patterns through one-way functions, and then store these test patterns as a public verification table. When logging, the submitted password is first encrypted as a test pattern and transmitted to the system. The system then verifies the received test pattern according to the corresponding entry in the verification table. This approach can protect users' passwords from being disclosed during login transmission.
- (2) Store the password in ciphertext form such that an intruder cannot easily derive the secret password even if the intruder knows the content of the verification table or can recognize the change in the verification table.

Nevertheless, the following shortcomings still exist in these enhanced password authentication schemes:

- (1) An intruder may encrypt his/her password by the one-way function, and then append it to the verification table. After this is done, the intruder can penetrate into the network with the forged password planted in the verification table.
- (2) If the content of the verification table is modified or destroyed by a malicious intruder, then the entire system may break down. In addition, the verification table needs extra memory space reserved in the system.
- (3) The verification table will be greatly expanded after many users have joined the network. The management of the password table then becomes increasingly complex.

Considering remote access systems, a potential impersonation problem exists. An intruder may intercept a valid login request (or an authentication message) and replay it later to pretend to be a legal user. In particular, this issue often occurs in a remote access system through insecure channels. In 1981, Lamport [9] proposed a remote password authentication scheme which can withstand such a replay attack, but the scheme is insecure if the encrypted passwords stored in the computer system is modified by intruders. Later, another scheme based on the signature scheme of a public key cryptosystem was proposed by Denning [2], but the scheme still need password tables. In 1991, Chang and Wu [10] used the Chinese remainder theorem to implement password authentication in a remote access network environment. In their method, once a user wishes to login to the network, he/she must submit his/her identity and password associated with the smart card. A smart card originates from an IC memory card used in industry about ten years ago [11, 12]. The main characteristics of a smart card are a small size and low power consumption. Generally,

a smart card contains a microprocessor which can quickly manipulate logical and mathematical operations, a RAM which is used as a data or instruction buffer, and a ROM which stores the user's secret key and the necessary public parameters and algorithmic descriptions of executing programs [5, 10, 13]. The merits of a smart card for password authentication are its high simplicity and efficiency of the login and authentication processes. However, in Chang and Wu's scheme users' passwords are generated by the password generation center, so it is inconvenient for a user to choose the password at his/her own wish. One year later, Chang and Lai [14] broke Chang and Wu's scheme. They assumed that the information stored on the smart card could be easily read out by a smart card holder. Using the public information of a smart card, the holder derives the secret decryption keys of the computer system. Therefore, the holder can find another user's password by intercepting the login transmission message. Recently, Chang and Hwang [15] proposed another remote password scheme without authentication tables to solve the remote password authentication problem. In their scheme, users are not allowed to freely select their own identities and passwords, and the system cannot easily delete a legal user, either. There is a common feature for the above schemes without authentication tables. The feature is that the system cannot delete users easily. If a password held by a user becomes unauthorized, then the system has to reconstruct all passwords for the users. Though the system does not keep any secret data, it also loses the autonomy for managing its users. This is not practical in our real world. Additionally, Wang and Chang's [16] and Wu's [13] schemes permitted users to freely choose the preferred passwords, but it cannot easily delete a legal user from the computer system, and cannot be efficiently adopted in multi-server Internet environments, either.

Based on smart cards, we propose a new remote login authentication scheme for multi-server Internet environments in this paper, which can verify a single password for logging multiple authorized servers without using any password verification table. For the environment of single server, the issue of remote login authentication has already been solved by a variety of schemes, but it has not been efficiently solved for multi-server Internet environments yet, where multi-server Internet environments may include FTP servers, file servers, database servers, or WWW servers. The objective of the new scheme emphasizes that any client can get service grant from multiple servers without repetitive registration to each server. The proposed scheme's advantages include that not only repetitive registration for various servers is avoided, but also the network users can freely choose their preferred passwords and be deleted easily by the system. We can show that in our scheme intruders cannot derive any secret information from the intercepted login transmission message. Moreover, security analyses about the impersonation and replay attacks on the proposed scheme also validate the feasibility of the scheme.

The rest of this paper is organized as follows. In Section 2, we depict the proposed smart card based remote login authentication scheme for multi-server Internet environments. Section 3 shows that the new proposed scheme can not only withstand both the impersonation and replay attacks, but also prevent intruders from obtaining any secret information by intercepting the login transmission message. Finally, some concluding remarks are included in the last section.

2 The Proposed Scheme

The proposed scheme can be divided into the following four stages: the system setup stage, the user registration stage, the login stage, and the server authentication stage. The central authority (CA) selects several public and secret system parameters in the system setup stage. In the user registration stage, the user chooses a password known only to himself/herself, and applies for the access privilege for each server. CA then delivers a smart card to the registered user. The smart card contains the user's identity and some necessary public and secret parameters used in the login and server authentication stages. When logging, the user first inserts his/her smart card into a terminal and keys in his/her password. The smart card then generates an authentication message and transmits it to each server. After receiving the user's login request (i.e. the authentication message sent from the smart card), each server can easily validate the login request by employing a verification equation derived from the authentication message.

[The system setup stage]

Initially, as in the RSA scheme [17], CA needs to select public keys N and e and secret keys p_1, p_2, d , and $\Phi(N)$ as follows:

- (1) Select two distinct large primes p_1 and p_2 , and calculate

$$N = p_1 * p_2 \text{ and}$$

$$\Phi(N) = (p_1 - 1) * (p_2 - 1),$$

where Φ is called the Euler totient function [2, 17].

- (2) Select a public key e to be an integer with $\gcd(e, \Phi(N)) = 1$.
- (3) Find a secret key d , the multiplicative inverse of e , such that

$$e * d = 1 \pmod{\Phi(N)}.$$

[The user registration stage]

Assume that a new user U_i registers to the multi-server system including m servers $S_1, S_2, \dots$, and S_m . Before U_i gets service grant from multiple servers, he/she must submit his/her application to CA in advance as below:

- (1) First choose his/her own identity U_ID_i and password U_PW_i , and present them to CA.
- (2) After verifying the qualification of U_i , CA issues the identity U_ID_i and the password U_PW_i to him/her.
- (3) CA computes U_i 's secret key

$$U\_R_i = g^{U\_PW_i * d}$$

where g is a fixed primitive element in the Galois field $GF(N)$ [18].

- (4) Suppose that U_i can get service grant only from servers $S_1, S_2, \dots$, and $S_r, 1 \leq r \leq m$, and the service period of these servers for U_i are $E\_T_{i1}, E\_T_{i2}, \dots$, and E_T_{ir} ,

respectively. Moreover, it is assumed that the service periods of other servers S_{r+1} , S_{r+2} , ..., and S_m in the system for U_i are all zero. CA then constructs a Lagrange interpolating polynomial $f_i(X)$ for U_i in the following:

$$\begin{aligned} f_i(X) &= \sum_{j=1}^m (U\_ID_i + E\_T_{ij}) \frac{(X - U\_ID_i)}{(S\_SK_j - U\_ID_i)} \prod_{k=1, k \neq j}^m \frac{(X - S\_SK_k)}{(S\_SK_j - S\_SK_k)} \\ &\quad + U\_R_i \prod_{y=1}^m \frac{(X - S\_SK_y)}{(U\_ID_i - S\_SK_y)} \pmod{N} \\ &= a_m X^m + a_{m-1} X^{m-1} + \dots + a_1 X + a_0 \pmod{N} \end{aligned}$$

where S_SK_j represents the secret key of j th server in the system.

- (5) CA stores the interpolating polynomial $f_i(X)$ in the secret data space of U_i 's smart card U_SC_i to prevent it from being disclosed or modified. In addition, CA stores U_ID_i and a one-way function $h(X, Y)$ in the protected data space of U_SC_i . Note that in our scheme, the secret data space of the smart card is for storing secret information, which is not allowed to read directly from the card and is used only for internal computations of the smart card. However, the data stored in the protected data space can be obtained by using the secret key.
- (6) CA delivers the smart card U_SC_i to U_i through a secure channel.

[The login stage]

When the registered user U_i logs in the server $S_j(1 \cdot j \cdot m)$, the following steps need to be performed by his/her smart card U_SC_i :

- (1) U_i first inserts his/her own smart card U_SC_i to a card reader and keys in his/her password U_PW_i . Afterwards, U_SC_i gets a timing sequence t from the system, where t is used as a timestamp for the login request.
- (2) U_SC_i generates a secret random number r_i and computes the two values C_1 and C_2 in the following:

$$\begin{aligned} C_1 &= g^{e * r_i} \pmod{N} \\ C_2 &= g^{U\_PW_i} * g^{r_i * h(C_1, t)} \pmod{N} \end{aligned}$$

- (3) Assume that S_ID_j is the identity of j th server, which can be derived from the following formula

$$S\_ID_j = g^{S\_SK_j} \pmod{N}$$

$$P = (S\_ID_j)^{e * r_i} \pmod{N} = (g^{S\_SK_j})^{e * r_i} \pmod{N} = g^{S\_SK_j * e * r_i} \pmod{N}$$

U_SC_i then computes the value P as follows:

- (4) Given 1, 2, ..., m , and P , U_SC_i calculates $f_i(1)$, $f_i(2)$, ..., $f_i(m)$, and $f_i(P)$.
- (5) U_SC_i constructs an authentication message $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \dots, f_i(m), f_i(P)\}$ and transmits it to the server S_j .

[The server authentication stage]

Let t_{now} represent the server S_j 's current date and time. After receiving the authentication message M sent from the login user U_i , S_j performs the following steps to authenticate U_i 's login request:

- (1) Validate the format of U_i 's identity U_ID_i . If it is invalid, then reject the login request.
- (2) Check whether t_{now} is too much later than the login timestamp t or not by examining if $t_{now} - t > \bullet T$, where $\bullet T$ is the endurable transmission delay between the login terminal and the server S_j . If $t_{now} - t > \bullet T$, then the authentication message M may be replayed by some malicious attacker, and thus the server authentication stage has to be terminated.
- (3) Find the value P by employing the value C_i and the secret key S_SK_j :

$$P = (C_i)^{S\_SK_j} \pmod{N} = (g^{e*r_i})^{S\_SK_j} \pmod{N} = g^{e*r_i*S\_SK_j} \pmod{N}$$

- (4) By using these $m + 1$ points $(1, f_i(1))$, $(2, f_i(2))$, ..., $(m, f_i(m))$, and $(P, f_i(P))$, reconstruct the original interpolating polynomial

$$f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \dots + a_1 X + a_0 \pmod{N}.$$

- (5) Find U_i 's secret key U_R_i by computing the following formula

$$f_i(U\_ID_i) = U\_R_i \pmod{N}.$$

- (6) Verify if the following equation holds by using C_1 , C_2 , and U_R_i :

(1)

$$\frac{C_2^e}{C_1^{h(C_1,t)} * (U\_R_i)^{e^2}} = 1 \pmod{N}$$

If the correspondence holds, then the login request is valid and S_j accepts the login user U_i ; otherwise, it rejects U_i . The following will demonstrate why the verification procedure described in Eq. (1) works correctly:

$$\begin{aligned} & \frac{C_2^e}{C_1^{h(C_1,t)} * (U\_R_i)^{e^2}} \pmod{N} \\ &= \frac{(g^{U\_PW_i} * g^{r_1 * h(C_1,t)})^e}{(g^{e*r_1})^{h(C_1,t)} * (g^{U\_PW_i * d})^{e^2}} \pmod{N} \\ &= \frac{(g^{r_1 * h(C_1,t)} * g^{e * U\_PW_i})}{g^{e*r_1 * h(C_1,t)} * (g^{e*d})^{e * U\_PW_i}} \pmod{N} \end{aligned}$$

$$\begin{aligned}
&= \frac{g^{e^*U - PW_i}}{(g^{e^*d})^{e^*U - PW_i}} \pmod{N} \\
&= \frac{g^{e^*U - PW_i}}{g^{e^*U - PW_i}} \pmod{N} \\
&= 1 \pmod{N}
\end{aligned}$$

Calculate $f_i(S\_SK_j)$ by using S_j 's secret key S_SK_j as below:

$$f_i(S\_SK_j) = U\_ID_i + E\_T_{ij} \pmod{N} \quad (2)$$

In order to know when the service period of S_j for U_i is, S_j can derive it from Eq. (2) because

$$E\_T_{ij} = f_i(S\_SK_j) - U\_ID_i \pmod{N}.$$

S_j then compares E_T_{ij} with its current date and time t_{now} to see if the service period expires. If $E\_T_{ij} \cdot t_{now}$, then S_j will terminate the service to U_i (i.e. U_i 's access privilege to S_j is deleted from now on).

3 Security Analysis

In this section, the security analyses of the proposed scheme are given. We will show that the proposed scheme can withstand the following possible attacks:

(Attack 1) The impersonation attack

An intruder may impersonate U_i by forging a valid authentication message $M' = \{U\_ID_i, t', C_1', C_2', f_i(1), f_i(2), \dots, f_i(m), f_i(P)\}$ and replaying it to the server. In order to pass the check of Eq. (1) in Step (6) of the server authentication stage, the intruder may calculate C_1' , C_2' and U_R_i' from the forged secret random number r_i' , U_i 's password U_PW_i' , login timestamp t' , and CA's secret key d' as follows:

$$\begin{aligned}
C_1' &= g^{e^*r_i'} \pmod{N} \\
C_2' &= g^{U\_PW_i'} * g^{r_i' * h(C_1', t')} \pmod{N} \\
U\_R_i' &= g^{U\_PW_i' * d'} \pmod{N}
\end{aligned}$$

However, as long as the forged d' is used in computing U_R_i' , Eq. (1) cannot be satisfied because our proposed scheme is constructed based on the security of the RSA scheme [17], which relies on the difficulty of factoring a large number into its prime factors. That is, if the intruder want to pass the verification of Eq. (1), then he/she must derive U_R_i' in the following form

$$U\_R_i' = g^{U\_PW_i' * d} \pmod{N}$$

But it is very difficult for the intruder to find d in our proposed RSA-based scheme. Consequently, the impersonation attack can be withstood successfully.

(Attack 2) The replay attack

Two possible replay attacks are considered in the following. First, an intruder intercepts an authentication message and tries to masquerade as the sender by replaying it without modifying any content of the authentication message; secondly, an intruder intercepts an authentication message and replays a forged authentication message modified from the original one. Since the authentication message is time-dependent, the first approach will be excluded from the check in Step (2) of the server authentication stage. To pass the check in Step (2) of the server authentication stage, the intruder must change the timestamp t to t' so that $t_{now} - t' > \Delta T$ holds. Once the timestamp t is changed, the value C_2 must also be changed such that the intruder can pass the verification of Eq. (1) at the server authentication stage. That is, the intruder must find C_2' to satisfy

$$C_2' = g^{U\_PW_i * r_1^{*h(C_1, t')}} \pmod{N}$$

However, based on the above analysis for the impersonation attack and the difficulty of solving the intractable discrete logarithm problem [18, 19], it is impossible for the intruder to successfully forge a valid C_2' to pass the server authentication without knowing U_i 's password U_PW_i . Hence, the second approach will also be excluded from the check in Eq. (1).

(Attack 3) Extending the authorized period of server service by users themselves

When S_j 's service period for U_i expires, he/she should resubmit his/her application to CA for extending the authorized period of server service if he/she wants to continue the server service. We are convinced that it is absolutely impossible for U_i to employ his/her original smart card to extend the service from S_j at the expiration of S_j 's service. As described in Step (4) of the user registration stage, each server S_j 's ($1 \leq j \leq m$) service period E_T_{ij} for U_i is hidden in the interpolating polynomial $f_i'(X)$ constructed by all servers' secret keys $S\_SK_1, S\_SK_2, \dots$, and S_SK_m . As long as U_i does not get the correct S_SK_j , he/she cannot absolutely construct a valid interpolating polynomial to pass the server authentication by himself/herself. In other words, U_i cannot succeed in passing the server authentication even though he/she attempts constructing $f_i'(X)$ to deceive S_j by finding m forged S_SK_j' . This is because he/she cannot pass the authentication in Eq. (1) owing to no ability to compute U_R_i such that $f_i'(U\_ID_i) = U\_R_i \pmod{N}$. Thus, the proposed scheme can prevent legal users from extending the authorized period of server service by themselves.

(Attack 4) Intercepting the authentication message to obtain secret data

In our proposed scheme, it is secure against the attack on intercepting U_i 's authentication message $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \dots, f_i(m), f_i(P)\}$ to obtain secret data. Based on the difficulty of solving the discrete logarithm problem, an intruder cannot derive the secret data r_i and U_PW_i from C_1 and C_2 . On the other hand, since the intruder cannot derive the value P without knowing

S_j 's secret key S_SK_j , he/she certainly cannot reconstruct the polynomial $f_i(X)$ of degree m to get servers' secret keys only from these m points $(1, f_i(1))$, $(2, f_i(2))$, ..., and $(m, f_i(m))$.

4 Conclusions

Password authentication is the most widely used authenticating technique because of its ease of implementation, user friendliness, and low cost. In this paper, we have proposed an efficient remote login authentication scheme for multi-server Internet environments using smart cards. The smart card plays an important role in our scheme. Using it together with a user's preferred password, the login request message is transmitted securely and the verification can be performed easily. For each legitimate network user, our proposed scheme can verify his/her single password for logging multiple authorized servers without constructing the password or verification table for authenticating login requests, and he/she can get service grant from multiple servers without repetitive registration to each server. Hence, we eliminate the threat of revealing the entire list of passwords, even if in ciphertext form, and obtain an efficient smart card based remote login authentication scheme for multi-server Internet environments.

In addition, in our scheme the system can delete illegal users easily and periodically. The privilege of each legal user for multi-server access is only validated before some date. When the authorized period is exceeded, the smart card held by a user will lose its authority automatically. This means that the corresponding user will become illegal from then on. Furthermore, it is also difficult that a legal user tries to extend his/her authorized period by himself/herself.

To ensure the security of sending the login request message over the public channel, the communication timestamp is provided in the authentication phase to withstand the potential attack of replaying a previously intercepted login request under the assumption of universally synchronized clocks. Besides, we have also shown that the proposed scheme is secure against the attack on forging authentication messages. Therefore, it is believed that our proposed remote password authentication for multi-server Internet environments is practical in real applications.

References

1. Davies, D.W., Price, W.L.: Security for Computer Networks -- An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer. 2nd Edition. Wiley, Chichester (1989)
2. Denning, D.E.R.: Cryptography and Data Security. Addison-Wesley, Reading, MA (1982)
3. Pfleeger, C.P.: Security in Computing. 2nd Edition. Prentice-Hall, Englewood Cliffs, NJ (1997)
4. Seberry, J., Pieprzyk, J.: Cryptography -- An Introduction to Computer Security. Prentice-Hall, Englewood Cliffs, NJ (1989)
5. Menkus, B.: Understanding the Use of Passwords. Computers & Security 7 (1988) 132-136

6. Purdy, G.P.: A High Security Login Procedure. *Communications of the ACM* 17 (1974) 442-445
7. Evans, A., Kantrowitz, W., Weiss, E.: A User Authentication Scheme Not Requiring Secrecy in the Computer. *Communications of the ACM* 17 (1974) 437-442
8. Lennon, R.E., Matyas, S.M., Meyer, C.H.: Cryptographic Authentication of Time-Invariant Quantities. *IEEE Transactions on Communications* 29 (1981) 773-777
9. Lamport, L.: Password Authentication with Insecure Communication. *Communications of the ACM* 24 (1981) 770-772
10. Chang, C.C., Wu, T.C.: Remote Password Authentication with Smart Card IEE Proceedings-E 138 (1991) 165-168
11. Peyret, P., Lisimaque, G., Chua, T.Y.: Smart Cards Provide Very High Security and Flexibility in Subscribers Management. *IEEE Transactions on Consumer Electronics* 36 (1990) 744-752
12. Sternglass, D.: The Future Is in the PC Cards. *IEEE Spectrum* 29 (1992) 46-50
13. Wu, T.C.: Remote Login Authentication Scheme Based on a Geometric Approach. *Computer Communications* 18 (1995) 959-963
14. Chang, C., Lai, C.S.: Comment on Remote Password Authentication with Smart Cards. *IEE Proceedings-E* 139 (1992) 372-372
15. Chang, C.C., Hwang, S.J.: Using Smart Cards to Authenticate Remote Passwords. *Computers & Mathematics with Applications* 26 (1993) 19-27
16. Wang, S.J., Chang, J.F.: Smart Card Based Secure Password Authentication Scheme. *Computers & Security* 15 (1996) 231-237
17. Rivest, R., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM* 21 (1978) 120-126
18. Diffie, W., Hellman, M.E.: New Directions in Cryptography. *IEEE Transactions on Information Theory* IT-22 (1976) 644-654
19. El-gamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory* 31 (1985) 469-472

NetLets: Measurement-Based Routing for End-to-End Performance over the Internet

Nageswara S.V. Rao^{1,3}, Sridhar Radhakrishnan², and Bang-Young Choi²

¹ Computer Science and Mathematics Division, Oak Ridge National Laboratory,
Oak Ridge, TN 37831, USA
raons@ornl.gov

² School of Computer Science, University of Oklahoma,
Norman, Oklahoma 73019, USA
sridhar@cs.ou.edu

Abstract. Routing in the Internet is based on the best-effort mechanism, wherein a router forwards a packet to minimize the number of hops to the destination. Also in the Internet, all packets are treated the same independent of their size. We propose the framework of NetLets to enable the applications to send data packets to the destination with certain guarantees on the end-to-end delay. NetLets employ built-in instruments to measure the bandwidths and propagation delays on the links, and compute the minimum end-to-end delay paths for data packets of various sizes. Based on our experiments, the paths selected by our system using the measurements are indeed the minimum end-to-end delay paths, and our method outperformed the best-effort mechanism based on the hop count.

1 Introduction

Routing is crucial in determining the end-to-end performance in the distributed computing applications over the Internet. Current routing protocols select the paths to minimize the number of hops to the destination. These protocols sometimes ignore viable alternate routes with higher bandwidths or less congestion, both of which are vital factors in deciding the end-to-end delays. Furthermore, one of the reasons for the unsatisfactory end-to-end path performance of the present Internet is the inadequate measurement infrastructure. Paxson *et. al* [8] point out the need for infrastructure to diagnose performance problems, measure the performance of network paths, and to assess performance of different Internet service providers. In this paper, we

³ Research of Rao is sponsored by Defense Advanced Research Projects Agency under MIPR No. K153, and by the Laboratory Director's Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC for the U. S. Department of Energy. In addition, research of Rao and Radhakrishnan is sponsored by the Engineering Research program of the Office of Science, U. S. Department of Energy.

employ the measurements but for the sole purpose of computing the minimum end-to-end delay paths.

Quality-of-Service (QoS) guarantees are currently needed by a variety for applications such as video conferencing and real-time distributed simulation experiments. To facilitate the growing number of such applications, new protocols and infrastructures are being developed (MLPS, DiffServe and IntServe [3,6]) for ensuring end-to-end performance. The real question is that, given the infrastructure currently in place, is it practical to replace it all with a new one that offers QoS guarantees for end-to-end performance? The answer is no at least in the near future, but there can be middle ground. This is achieved by incorporating daemons, called the *NetLets*, above the existing network layer to provide end-to-end performances beyond what is possible in the current Internet. NetLets will be deployed on certain intermediate routers (if possible) and on host machines. Two NetLets are connected via a *virtual link* if they are connected by an IP (Internet Protocol) path containing zero or more routers of the underlying network. In Figure 1 an example of the virtual network that overlays on a real network is shown.

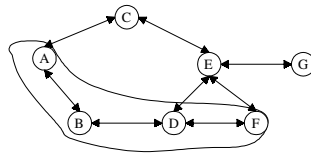


Fig. 1: The shaded nodes house NetLets. A virtual link from node A to node F corresponds to the path A-B-D-F in the underlying network.

We now show that the packet size has a strong influence on the end-to-end delay of routing paths. It is indeed possible for the same source and destination pairs to have different minimum end-to-end delay paths for packets of different size. Consequently, the routing policy based on minimizing the hop count is only sub-optimal in general. The size of an IP packet can vary from a minimum of 21 bytes (20 bytes of header and 1 byte of data) to a maximum of 65,535 bytes (20 bytes of header and 65,515 bytes of data). Consider the network shown in Figure 2. The nodes A and D are the source and destination, respectively. The bandwidths on the links have been chosen to be either DS1 or DS1C and all links are assumed to have a propagation delay of 10μsecs. Then, to a first order of approximation, the end-to-end delay of a

link is proportional to $\frac{\sigma}{B} + d$, where σ is the size of the data packet, B is the

bandwidth on the link and d is the propagation delay of the link. In the table below the network diagram, the end-to-end delay of the routes for A to D for various packet sizes are shown. Clearly, for the minimum sized IP packet one should choose the path A-B-D, while for the maximum sized IP packet one should choose the path A-B-C-D. At the intermediate nodes, we have not included the queuing and processing delays, collectively called the *node delays*. For the maximum sized IP packet, we

choose A-B-D if the node delay at C is greater than 7msecs. The node delays are often much harder to estimate unlike the bandwidth and link delays. These delays can be handled by utilizing the probabilistic components to the end-to-end delays to derive similar conclusions but with a certain probability [11].

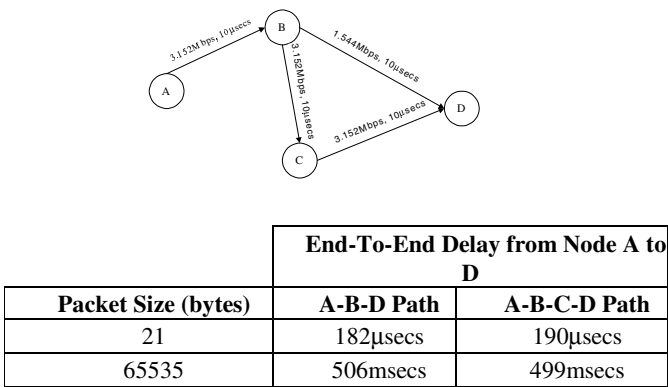


Fig. 2. A network with A and D as the source and destination, respectively. The bandwidth and propagation delays on the links are indicated. The table indicates the end-to-end delays for minimum and maximum sized IP packet using different paths.

As evident from the above example, protocols that ignore the available bandwidths and propagation delays will produce routing paths with sub-optimal end-to-end performance in certain cases. We present a framework that accounts for the bandwidth and propagation delays:

1. software instruments are used to measure the bandwidths and propagation delays on the links;
2. bandwidth and delay estimates are used to determine the minimum end-to-end delay paths; and
3. interfaces are provided for the applications to route data packets along the computed paths.

Analytical basis for this approach is first provided in [11], where it was shown that measurements are sufficient (with a specified probability) to compute the paths with the minimum end-to-end delay, irrespective of delay distributions. It is important to note that it is not necessary to derive delay distributions for providing this type of end-to-end QoS, and one can achieve useful performances based on measurements alone.

We compare the paths chosen by our technique with those chosen by the best-effort mechanism in an experimental setup of workstations that route data packets using the Internet Protocol. In the experiments, the paths selected by our system based on the measurements indeed have the minimum end-to-end delay. Our method outperformed the best-effort mechanism and in this sense, our results provide an experimental evidence for the analytical results in [11,15].

2 Related Research

Research on the end-to-end performance of path section protocols has received much attention in recent years. For example, Savage *et. al.* [17] in their Detour system point out various performance problems in the Internet including the inefficiencies both in the routing and transport layer protocols. Collins [5] also suggests the formation of a virtual network similar to the one proposed in this paper. There are many differences between our work and the Detour system. First, the Detour system does not explicitly minimize end-to-end delay between source and destination pairs. Second, the measurements mechanism suggested by the Detour system involves the use of utilities such as *traceroute* and *ping*. Our measurements (link bandwidth and link propagation delay) are obtained using the actual transport layer segments, namely TCP (Transport Control Protocol) segments. That is, the measurements are done using transport layer units that are actually routed on the network rather than control data packets sent via *traceroute* and *ping* programs. With the deployment of firewalls, the Internet Control Message Protocol (ICMP) traffic might be treated significantly different from TCP or User Datagram Protocol (UDP) traffic; for example, some firewalls disable responses to *ping* and provide no or misleading responses to *traceroute*. The TCP-based measurement scheme used here is vital to proving the analytical guarantees of [11,15], and no such guarantees can be given if only *traceroute* and *ping* measurements are used. Third, unlike the Detour system we do not modify the existing IP layer but direct the IP layer to forward the packets via the other NetLets.

Paxson [7] has developed measurement tools for determining the end-to-end delay, packet loss, and actual routing paths on the Internet. The RFC 2330 [8] examines the framework for measuring IP performance metrics. Most of this research points out the sensitivity of the measurements with respect to the changes in clock. For example, in a time synchronized distributed system, the one-way link time can be measured by time-stamping a packet before sending and time-stamping it after it reaches the destination. Since such assumptions are non-plausible in a distributed system, Paxson [7] suggested techniques to adopt to arrive at a reasonably accurate packet delay measurements. While measurements in general are motivated by several purposes, such as traffic modeling and fault diagnosis, the goal of our measurements is limited to reducing the end-to-end delays.

IP routing involves designating groups of nodes (routers and hosts) to form an Autonomous System (AS). The entire system of routers and hosts are grouped into different ASs, and communication between any two nodes within an AS is carried out using the Interior Gateway Protocol (IGP). The Border Gateway Protocol (BGP) treats each AS as a single node and provides the routing protocols between two nodes that are in different ASs. Both IGP and BGP use the hop count as a performance measure to determine end-to-end path [9], that is, given two paths between a source and destination pair, the path with fewer links will be chosen.

To transfer messages, there are two basic routing mechanisms: circuit switching (also called *pipelining*) and packet switching based on the *store-and-forward* method. When data is delivered using circuit switching, bit streams of data are transferred at a fixed rate from the source to destination without buffering. Over the packet switched

networks, entire data is stored at every intermediate node before being forwarded to next node. The telephone networks belong to circuit switching, and the IP networks belong to packet switching paradigm. For path P , the *path-delay* $D(P)$ is the sum of all propagation delays of links along the path. The *end-to-end delay* of a path P can be computed by the formula $T = \sigma / B(P) + D(P)$ in the circuit switching, where $B(P)$ is the *path-bandwidth* which is appropriately computed based on the routing mechanism. Since in circuit switching the data transfers along the route is with a fixed rate, $B(P)$ is the minimum bandwidth of link along the path. In the case of packet switching, since the incoming data is stored temporarily at each node and then transmitted to an outgoing link, transmission time of $\sigma / B(e)$ is required at each node v , where e is the outgoing link of v . Thus, for packet switching mechanism, the path-bandwidth $B(P)$ is $\frac{1}{\sum_{e \in P} \frac{1}{B(e)}}$, where P is the routing path and e is the link on P [14].

In summary, the routing mechanism has a significant effect on the end-to-end delay and must be considered in computing the optimal end-to-end delay paths.

The well-known *quickest path problem* is to find a routing path in a network G such that the end-to-end delay time required to send σ units of message from a source to a destination is minimum. Chen and Chin [4], Rosen *et. al.* [16], Rao and Batsell [10-12], and Bang *et. al.* [2] studied the quickest path problem using the circuit-switching mode. However, since the store-and-forward transfer mode is used to send message in the Internet, the classical quickest path algorithm cannot be adapted to an IP network. In this paper, we take into account bandwidth and propagation delay of links to minimize the end-to-end delay of the path in an IP network. We compare the method based on minimum number of hops with our method based on minimum end-to-end delay by utilizing the estimated bandwidths and propagation times of links.

3 Estimation of Bandwidth and Propagation Delay

We measure the bandwidth of a link (a, b) , which can be a virtual link or a physical link in the underlying network, using the following steps:

1. Generate various sizes for a TCP segment at node a ;
2. For each size s , construct a number of TCP segments of size s and send them to node b .
3. Node b upon receiving each TCP segment simply echoes it back to node a .
4. The round-trip delay is measured for each segment and the average is computed. The end-to-end delay is the round-trip delay divided by two.

Once the average end-to-end delay is determined for messages of different sizes,

and the linear regression is applied to determine the line $\frac{\sigma}{B} + d$ that fits the points as shown in Figure 3. From this computation, we obtain the “effective bandwidth” B of

the link. To determine the propagation delay (the time for a minimum size message to travel along the link), we sent a minimum size message containing 21 bytes several times and averaged it.

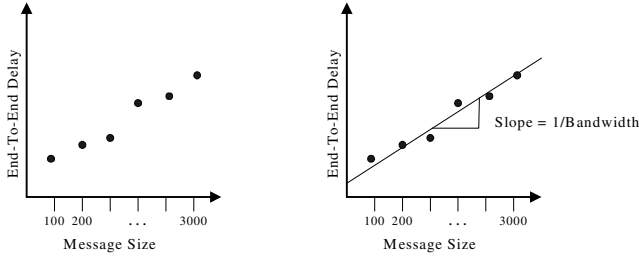


Fig. 3. Determination of the end-to-end delay experienced by transport layer segments of various sizes.

4 NetLets Framework

NetLets provide a software interface that the applications use to route messages via the minimum end-to-end paths (more details can be found in [15] on the overall framework of NetLets). Based on the bandwidth B and the propagation delay D of each link in the virtual network, the minimum end-to-end delay path for a message of particular size σ is determined by assigning to each link the weight $\frac{\sigma}{B} + d$, and

computing the shortest path in the resulting network using the one-to-all Dijkstra's shortest path algorithm [9,13]. Clearly, given any source and a set of messages of sizes $\sigma_1, \sigma_2, \dots, \sigma_k$, we can construct the shortest path trees with respect to each message size and if two trees are the same we can eliminate one of them. Once the shortest path trees are known, the routing tables are constructed. We perform this operation for all network nodes that house NetLets.

The routing table contains for each destination the next hop IP address of the node containing the NetLets modules. All NetLets modules communicate using predetermined port numbers. The NetLet after receiving a message from the local application sends the datagram using the designated port number and the IP address in the routing table. NetLet upon receiving the message determines if the packet is destined for it or for any nodes that it can route and performs the necessary actions.

5 Experimental Results

For experimentation we selected two virtual network topologies as shown in Figure 4 (a) and Figure 4 (b). As evident in topology 1 (Figure 4(a)), there are four different paths from source to destination node with the minimum number of hops, namely 4. Using the link weight of $(\sigma / B + D)$ for each link, the minimum end-to-end delay

path from source L to the destination N is $L \rightarrow J \rightarrow T \rightarrow N$. Also, the paths, $L \rightarrow J \rightarrow T \rightarrow N$, $L \rightarrow I \rightarrow T \rightarrow N$, $L \rightarrow J \rightarrow V \rightarrow N$, and $L \rightarrow I \rightarrow V \rightarrow N$, are paths with minimum number of hops. For topology 2 (Figure 4(b)), paths with minimum number of hops are $I \rightarrow J \rightarrow T \rightarrow N$, $I \rightarrow J \rightarrow V \rightarrow N$, and $I \rightarrow L \rightarrow T \rightarrow N$, where I and N are source and destination, respectively. The minimum delay path is $I \rightarrow J \rightarrow T \rightarrow N$ which also has the minimum number of hops.

With respect to the network topology 1, Figure 5(a) and 6(a) represent the observed end-to-end delay for paths computed by sending messages of different sizes, and Figure 5(b) and 6(b) show the end-to-end delay obtained by our method (namely, the summation of the link weights on each of the paths). Even though the observed end-to-end delay is different in magnitude from the calculated end-to-end delay, the two methods have chosen the same path from source to destination as the minimum end-to-end delay path.

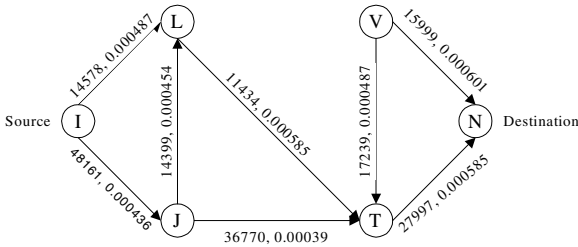


Fig. 4(a). Virtual topology 1 containing six nodes with bandwidth in bytes per second and propagation delay in seconds indicated on the links.

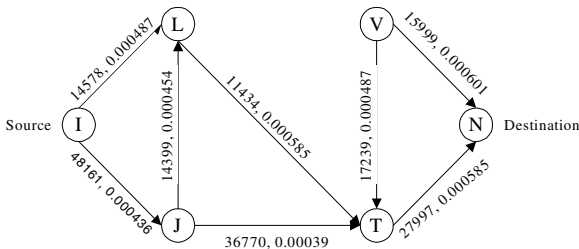


Fig. 4(b). Virtual topology 2 containing six nodes with bandwidth in bytes per second and propagation delay in seconds indicated on the links.

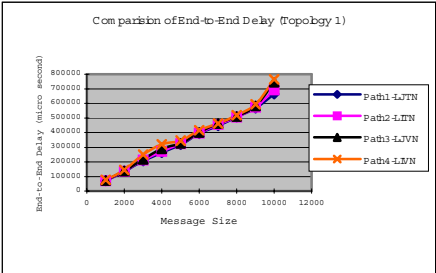


Fig. 5(a). For virtual topology 1 this graph contains the observed end-to-end delay for various paths.

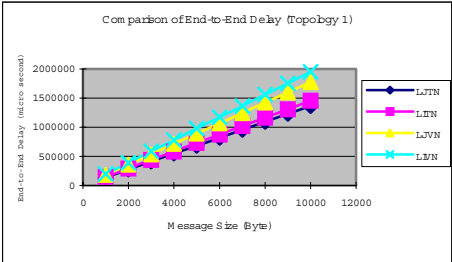


Fig. 5(b). For virtual topology 1 this graph contains the calculated end-to-end delay for various paths from source to destination.

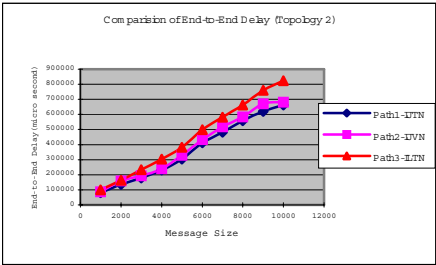


Fig. 6(a). For virtual topology 2 this graph contains the observed end-to-end delay for various paths.

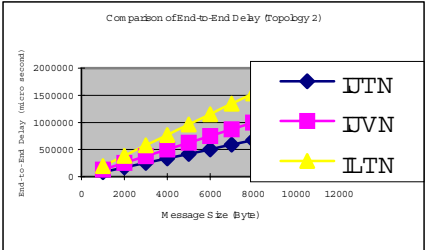


Fig. 6(b). For virtual topology 2 this graph contains the calculated end-to-end delay for various paths from source to destination.

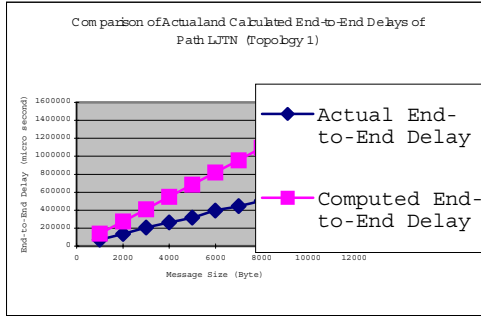


Fig. 7. Comparison of observed (actual) and computed end-to-end delay of the path LJTN in topology 1.

In Figure 7 we show the observed and computed end-to-end delays. While there are clear differences in delay times, the minimum end-to-end delay paths chosen by computation and experimentation are exactly the same path.

6 Conclusions

We presented experimental results on the performances of the routing paths, comparing the minimum hops method with our method that computes minimum end-to-end delay paths based on measurements. Our results showed that the minimum delay path computed based on the estimated bandwidth and delays outperformed the path with minimum number of hops. This is because the end-to-end delay depends on the message size, propagation delay and bandwidth. We also proposed a software framework based on NetLets [14] that can be used for routing with probabilistically guaranteed performance.

Future work could involve utilizing other regression estimation methods [11,15]. Also, in terms of experimentation using Internet, NetLets were used in [14] for realizing two-paths between nodes that are geographically separated by thousands of miles. Our system can be expanded to incorporate more nodes and also the nodes that are more widely distributed over the Internet. NetLets are complementary and upward-compatible with QoS mechanisms such as MPLS, DiffServe, and IntServe as well as adapted and/or optimized versions of transport mechanisms such as auto-tuned TCP[18]. It would be interesting to see if existing NetLets can be enhanced to exploit these mechanisms to provide end-to-end performance beyond what is possible in current IP networks. Also, active networks [1] can be exploited to provide more information and control to the NetLets by attaching measurement and routing codes to the messages.

References

1. Active networks, <http://www.sds.lcs.mit.edu/activeware>
2. Y. C. Bang, S. Radhakrishnan, N. S. V. Rao, and S. G. Batsell, "On update algorithms for quickest paths," *Computer Communications*, vol. 23, no. 11, pp. 1064-1068, 2000.
3. B. E. Carpenter and D. D. Kandlur, Diversifying Internet delivery, *IEEE Spectrum*, Nov 1999, pp. 57-61.
4. Y. L. Chen and Y. H. Chin, "The quickest path problem," *Computers and Operations Research*, vol. 17 no. 2, pp. 153-161, 1990.
5. A. Collins, "The Detour Framework for Packet Rerouting," Technical Report, University of Washington, October 1998.
6. A. Dutta-Roy, The cost of quality in Internet-style networks, *IEEE Spectrum*, September 2000, pp. 57-62.
7. V. Paxson, "End-to-End Routing Behavior in the Internet," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 601-615, October 1997.
8. V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP Performance Metrics," *Network Working Group, RFC 2330*, May 1998.
9. L. Peterson and B. Davie, "Computer Networks: A Systems Approach," Second Edition, Morgan Kaufmann Publishers, 2000.
10. N. S. V. Rao and S. G. Batsell, "Algorithm for minimum end-to-end delay paths," *IEEE Communication Letters*, vol. 1, no. 5, pp. 152-154, 1997.
11. N. S. V. Rao and S. G. Batsell, "On routing algorithms with end-to-end delay guarantees," in the proceedings of *IC3N: International Conference on Computer Communications and Networks*, pp. 162-167, 1998.
12. N. S. V. Rao and S. G. Batsell, "QoS routing via multiple paths using bandwidth reservation," in the proceedings of *IEEE INFOCOM98: The Conference on Computer Communications*, volume 1, pp. 11-18, 1998.
13. N. S. V. Rao, W. C. Grimmell, S. Radhakrishnan and Y. C. Bang, Quickest paths for different network router mechanisms, Tech. Rep. ORNL-TM-2000/208, Oak Ridge National Laboratory, June 2000.
14. N. S. V. Rao, NetLets: End-to-end QoS mechanisms for distributed computing in wide-area networks using two-paths, <http://saturn.epm.ornl.gov/~nr Rao>.
15. N. S. V. Rao, End-to-end delay guarantees in computer networks: Analysis and NetLet implementation, ORNL Draft, <http://saturn.epm.ornl.gov/~nr Rao>.
16. J. B. Rosen, S. Z. Sun, and G. L. Xue, "Algorithms for the quickest path problem and the enumeration of quickest paths," *Computers and Operations Research*, vol. 18, no. 6, pp. 579-584, 1991.
17. S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan, "Detour: Informed Internet Routing and Transport," *IEEE Micro*, vol. 19. no. 1, pp. 50-59, January 1999.
18. Web100 Concept Paper, <http://www.web100.org>

Using the Internet in Transport Logistics – The Example of a Track & Trace System

Kai Jakobs, Carsten Pils, and Michael Wallbaum

Technical University of Aachen, Computer Science Department, Informatik IV
Ahornstr. 55, D-52074 Aachen, Germany
{Jakobs, Wallbaum, Pils}@i4.informatik.rwth-aachen.de

Abstract. The paper presents some socio-economic and technical issues that arise from the design of an open, multimodal end-to-end tracking & tracing system. An outline of the general background and the project's goals is followed by a discussion of the associated socio-economic aspects. User requirements are also addressed, with respect to the business context of the logistics environment. Subsequently, a brief overview of the project's technical design choices and is given, followed by a more detailed discussion of the individual components of this architecture.

1 Background and Goals

Europe has the most advanced transport infrastructure of the world, including a very dense road network, a modern rail network and an air network that covers the entire continent. Europe also has very advanced communication networks, consisting of dense, high-quality fixed networks, satellite coverage, and wireless communication networks, such as GSM and – in the near future – GPRS (General Packet Radio Service) and UMTS (Universal Mobile Telecommunication System). Unique opportunities will be generated when these two infrastructures are connected to each other.

Transport and logistics today have evolved into a high-technology industry. Distribution is no longer about moving cargo over road or via air from A to B, but is a complex process based on intelligent systems for sorting, planning, routing, and consolidation that supports faster transportation, different transportation modes, fallback scenarios in case of failures, value added services such as time sensitive deliveries and tracing of products throughout the supply chain or transport network.

Many large logistics companies have developed solutions for delivering these services in order to meet the requirements of their customers and to improve their services. Smaller companies, however, cannot afford these investments and are mainly active in the 'old' point-to-point transportation market, or co-operate with the larger companies, using their respective systems.

The companies that have the necessary information systems in place to participate in the market for high-end transport solutions normally offer their customers methods for tracking and tracing their consignments. Even though many customers would benefit from using this information in their own information systems, only few of them are doing this today because of the large investments in their systems required to

adapt to the proprietary interfaces of the transport companies. However, these systems typically have two major drawbacks:

- They do not normally work across company boundaries.
- They do not provide accurate ‘life’ information about location and, particularly, the status of individual units or items.

That is, continuous information about the current position or status of transport goods (in the sense that the geographic position can be queried at any time) is not commonly available today. Existing solutions are typically based on scanning bar codes at process or control points. Moreover, information at item level is not normally available either. Typically, this information is provided – if at all – at a vehicle or container level only. Furthermore, very few companies have true global or even European coverage. In daily business, products are frequently shipped by subcontractors of the transport company, which frequently means that tracking and tracing at least becomes much more difficult, and typically will no longer be possible at all. Only in a few cases do carriers exchange tracing information, but in most cases the costs for adapting the proprietary systems to each other are prohibitive.

The key idea of the ParcelCall project is to provide relevant services on top of TCP. This allows considerable freedom with respect to the underlying network protocols, potentially including e.g., GSM/GPRS, ISDN, UMTS, and IP. Easy integration into legacy systems, operated by the individual carriers, to the new information infrastructure is another key design criterion. Seamless interoperation between these systems on the one hand and the new tracking & tracing system has to be guaranteed.

Furthermore, and in addition to the track & trace services typically provided today, ParcelCall will enable [1]:

- item level tracking & tracing

That is, the granularity of the track & trace function will be adjustable, from the individual item (e.g., a parcel) up to a container or a vehicle,

- ‘near-real-time’ (continuous) tracking

Status and position information will be made available at anytime (not just at e.g. terminals or hubs).

The remainder of the paper is organised as follows: Chapter 2 briefly introduces the socio-economic context of the project. An overview of the user requirements from a business perspective is given in chapter 3. Subsequently, chapter 4 introduces the overall system architecture and its individual elements. Finally, some conclusions are presented in chapter 5.

2 The Socio-economic Context

The socio-economic environment within which freight forwarders and logistics companies operate is briefly outlined in this chapter.

The success of a new technology depends on more than simply its technical efficacy; it must also be matched with its socio-economic context. In some cases this means tailoring technology to the existing environment, in others the market and context may need to be ‘created’, alongside the technology, by the technology’s developers. Most obviously a technology must address the requirements of its various

users. Typically, it is important that current needs, as seen in existing business practices, are taken into account. However, although existing practices provide a starting point, gaining the full benefit of new technology often depends on its more radical application.

Tracking & tracing systems need to address the requirements of two main types of transportation: business-to-business and business-to-consumer. While the former increasingly hinges on efficient logistics management, a key issue for the latter, especially as regards the growth of e-commerce, is customer satisfaction.

High quality tracking and tracing of parcels matters for business-to-business transportation because of the trend towards inventory reduction. The speed, reliability and timeliness of delivery have increasing commercial salience both in procurement and in the quality of service offered by a supplier. Enhanced logistics management based on Just-in-Time, Vendor Managed Inventory or similar approaches can not only minimise stocks held, but can also involve outsourcing logistics management either to the supplier or to a specialist logistics operator. With e-commerce, boundaries between different 'stages' in the supply chain may become eroded. Distributors may take on extended roles; for example, in fulfilment and final assembly.

Improvements in tracking and tracing can also play a significant part in eliminating one of the problems faced by Internet shopping – reliable, time-assured delivery, tailored to customer requirements. Although not unique to Internet shopping, heightened customer expectations along with internet/WAP access provide an opportunity for improved customer service using more accurate parcel tracking technology.

While fulfilling these business applications is central, it is also important to recognise that there are a variety of other socio-economic issues that may affect the technology's success. Security and confidentiality may be important. Above all, a technology that involves inter-organisational data exchange depends heavily on the success of standardisation efforts and on the willingness of firms to work together. These issues may affect the technical choices adopted in the design and configuration, as well as the commercial strategies for its promotion. Strategic thinking on these lines is embedded in the architecture and strategy of the ParcelCall project [2], [3].

3 User Requirements

The user requirements discussed in this section relate rather more to the underlying business processes than to a technical architecture or a specific realisation.

Attaining technical objectives will be of little significance if the technology itself is not widely implemented. Although individual companies could benefit from its local adoption, a system's full potential lies in the development of a standardised approach that can gain general acceptance in the industry. Success will not depend simply on the development of the 'best' technology; equally important is the development of a constituency of users. The system will depend upon aligning expectations to ensure that a sufficient number of key users (critical mass) will be convinced to take part. It is crucial to convey that this represents the way forward, to win these kinds of commitments.

Thus, it is crucially important to recognise the diversity of players involved, with their very different commitments and needs. The development of a new Inter-

Organisational Network System may involve an uneven distribution of costs and benefits between these players [4]. In particular, it is important to ensure low barriers to entry - particularly for those players for whom a sophisticated track & trace system does not offer significant immediate benefits or strategic importance.

System senders, receivers, and carriers are the main users of a tracking & tracing systems. Their respective requirements are discussed in this section. Other users include Transport Broker, Packaging Services, Collection and Delivery Services, Depot/Hub/Terminal operators, and Vehicle Drivers.

Individual senders will typically take the parcel to a collection office. Home collection could also be possible, and it should be possible to arrange this service though Internet/WAP access. The sender would like the options of email confirmation of parcel delivery, and Internet/WAP access to transit status and estimated time of arrival.

The requirements of company senders will vary according to their business practices, and companies that are supplying goods to individuals may have similar requirements to individual senders. They will want to receive as much status information as possible because this can then be provided to their customers, providing value added to their service. Likewise, reliability in delivery times (or flexibility in rearranging them) is important, as would be the ability to confirm that the parcel has been received by the appropriate individual.

Individual receivers want to know when a parcel will arrive so that they can ensure that someone is there to receive it. Internet/WAP access and email messages can provide an attractive customer service, and is, for example, likely to be an important aspect of the development of internet shopping. This service could include 'real time' information on the parcel's movement, with updates in the estimated time of arrival being the key feature.

In the case of corporate receivers, their dealings with senders will often be part of long-term supply relationships. For example, in B2B e-commerce, many manufacturing receivers may use EDI to send orders or call-offs based on long-term supply agreements and these may be generated directly from their internal systems. ID tags should have the potential to satisfy any of the receivers' internal requirements for tracking the parcel.

Companies providing express delivery, freight-forwarding and logistics management will be the main users. Their customers (senders and receivers) may have certain data requirements (as noted above), but mainly they will want a high level of performance and service to be provided to them seamlessly and transparently. While a number of large, integrated carriers mainly use their own transportation (planes, trucks, etc), even they typically need to subcontract the physical carriage some of the time.

Parcels must carry an ID tag. As with existing track & trace systems, this tag will be read at each control point – typically the hand-over between transportation units when arriving at or leaving a depot. With active tags, the reading of tags can be continuous, or at customer request rather than solely at handover points.

The carrier also requires regular feedback from moving transport units where this is possible. This information will comprise location and transport unit status (is it on time? what is the expected delay?), along with all the parcels carried and their routing, destination and estimated time of arrival.

Delays or route deviations will be identified by the system. It should also be possible for deviations to be manually notified by the transport operator. If 'thinking

tags’ are used, then any undesirable deviations in the status of the parcel (in temperature, for example) should result in alerts to both the carrier’s main system and directly to the transport unit operator so that remedial action may be taken as soon as possible. Issues arise about whether to standardise these messages and, if proprietary and encrypted messages are being transmitted, whether carriers will feel happy to pass on to third parties information that they cannot themselves understand.

4 The Architecture

This chapter discusses the system architecture (see Figure 1). An overview of the specific architecture designed by, and deployed within, the ParcelCall project is followed by a description of the individual components.

A Mobile Logistics Server (MLS), located on board a vehicle, collects information on the individual items, including position and status. The former is obtained via the Global Positioning System (GPS), ‘intelligent’ tags are utilised to collect the latter. These ‘Thinking Tags’, have also been developed within the project, can form ad-hoc networks that can be applied to self-adapting hierarchical packing schemes or to active status monitoring of critical freight contents. Alarm messages will be actively generated if, e. g., an item enters a critical state (temperature, humidity, pressure, acceleration, etc.).

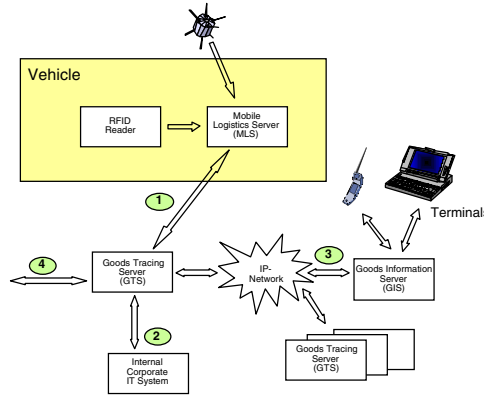


Fig. 1: The System Architecture

The MLS sends the compiled information to a Goods Tracing Server (GTS). Every participating company (e.g., freight forwarders, logistics service providers, fleet operators, etc) needs to install at least one GTS which also serves as the interface between the respective internal IT system and the track & trace service. Thus, the set of GTSs forms a highly distributed data base holding the information available to the end-users (subject, of course, to appropriate access rights and successful authentication). The individual servers are interconnected via public networks (as e.g. the Internet or ISDN). It should be noted that even very small companies which do not have their own tracking and tracing system can utilise the ParcelCall service, as a GTS (typically a PC), one (or a few) MLSSs, and some ‘thinking tags’ are pretty much the only additional pieces of hardware required. Customers can access the system

through a ‘Goods Information Server’ (GIS), whose tasks also include authentication and access control. The elements of this architecture are discussed below in more detail.

4.1 The Mobile Logistic Server

To provide the required services each transport unit need to be equipped with a Mobile Logistic Server (MLS; see Figure 2) which keeps track of the goods within that unit. A transport unit may, for instance, be a truck, a freight wagon, or a container.

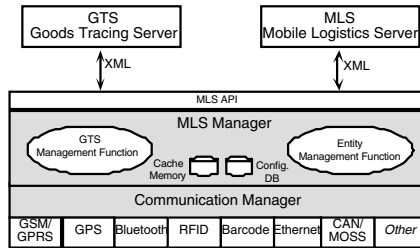


Fig. 2: Structure of an Mobile Logistic Server

Each such unit may contain transport goods or other units, thus potentially forming a hierarchy of transport units. Since each unit might have a MLS, the transport unit hierarchy causes an equivalent MLS hierarchy. Except for the top level MLS (root of the hierarchy tree) all MLSs communicate with their respective superior and subordinate MLSs.

Except for the top level MLS (root of the hierarchy tree) all MLSSs communicate with their respective superior and subordinate MLSSs.

From an MLS's point of view there is no difference between a tag and an MLS. A 'normal' MLS only needs to implement an item¹ interface. Only a top level MLS must implement an item interface and a GTS interface.

Both ‘thinking tags’ and MLSs store a unique item identifier, the item’s destination address, and certain other information, as e.g., constraints on e.g. temperature, shocks, or humidity. If a threshold of one of these constraints is exceeded an event will be generated and passed to an superior MLS, which forwards the event to its associated GTS. The GTS, in turn, forwards the event to the carrier’s IT system (which may or may not react upon the event). If a transport unit within this message chain has a control system (e.g., for a refrigeration unit), this control system can register with its associated MLS to receive events. Thus, transport units can also react immediately to certain events. This is of crucial importance as a transport unit, for instance a vehicle, might be disconnected from the carrier’s IT system (due to, e.g., poor GSM coverage) and thus cannot receive any instructions.

Unloading and re-loading of transport units changes the MLS hierarchy. To keep track of such changes items are scanned while being loaded. As the carrier's IT system plans the loading process in advance the ParcelCall system can check the

¹ Both goods and transport units within a container are subsequently referred to as ‘items’.

loading procedure while scanning the items. Therefore, each MLS can receive a loading list of ‘its’ items in advance from a GTS. While an item is scanned the MLS checks whether or not it is on its loading list and generates an alarm message if a ‘wrong’ item has been detected. Likewise, the responsible GTS is informed when the loading procedure has been completed successfully.

4.2 The Goods Tracing Server

The GTS network forms the backbone of the ParcelCall system (see Figure 3). It interconnects and integrates the individual ParcelCall servers on the one hand, and the individual carriers’ IT systems on the other. A GTS comprises of two databases: A Home Database and a Transitory Database. The Home Database contains item information, e.g. an item’s current position, its expected time of delivery, etc. The Transitory Database holds the MLS hierarchies.

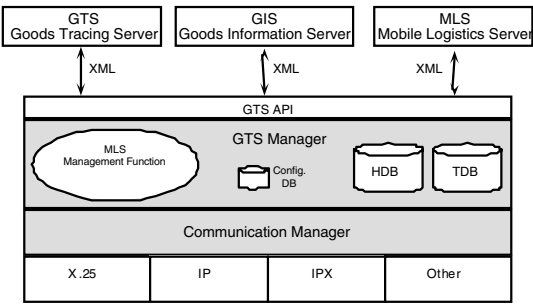


Fig. 3: GTS Internal Architecture

The Home Database. As the ParcelCall system must scale it is impossible to store information of each item within each Home Database. A GTS is responsible for a fixed number of top level MLSs (top level MLS cannot move). When an item enters the ParcelCall system for the first time it must be either registered with the GTS or with an MLS. The latter forwards registration information to the local GTS, which stores the item status information and the identification of the responsible top level MLS in its Home Database. Note that the top level MLS which is currently responsible for that item might belong to another GTS. In that case it is also necessary to store the identification of this GTS. Thus, the Home Database contains information of those items which have entered the system within the responsibility of the GTS.

A GTS receiving information about a certain item which is not registered at its Home Database forwards this information to the responsible GTS (the ID of which is stored in the tag). Having both the unique ID of an item and the ID of the responsible GTS it is straightforward to retrieve information about this transport good.

The Transient Database. It contains several MLS hierarchies (one for each top level MLS). To request instant status information of a certain item, the request must be routed through the MLS hierarchy. Having obtained the ID of the responsible top

level MLS it is straightforward to retrieve the required routing information from the Transitory Database. Note that the Transitory Database has one entry for each item which is currently under control of the GTS.

The Carrier’s IT System. Among others, the GTS has an interface to a carrier’s IT system. This system provides delivery plans to the ParcelCall system which are used to establish the MLS hierarchy. On the other hand, the GTS forwards status information and alarm messages to the carrier’s system.

4.3 The Goods Information Server

The Goods Information Server (GIS; Figure 4) provides customers with status information about their transport goods. To this end a GIS connects to a carrier’s GTS to retrieve this information. Basically, the GIS displays the information received from the GTS network to the user. Therefore, it implements a multimedia converter which allows conversion of different. The GIS also performs user authentication and manages the access control.

To retrieve information about a certain transport good a user authenticates to the GIS and provides the ID of the good as well as the ID of the responsible Home Database. Having both IDs it is straightforward to retrieve the information from the GTS network.

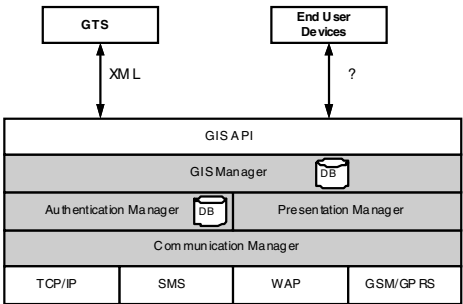


Fig. 4: The GIS Architecture

4.4 Passive and Active Tags

Passive RFID (Radio Frequency Identification) tags are available today at moderate costs, and can easily be integrated into labels holding e.g. bar code information. Due to the costs (compared to simple printed tags) and infrastructure requirements (printers, readers), this technology has yet to gain widespread acceptance for tagging of short life-cycle products and low value transactions. However, we believe that it is only a matter of a few years before RFID tags will play an active role in global markets. Static RFID tags – with limited data capacity and read-only access – can already be printed using standard printers with special ink, without the need of integrating any hardware (chips). RFID tags with read/write capability and a capacity of a few hundred bytes are available as low-cost one-chip solutions.

Complementing bar code labels with RFID tags will enable automatic identification ‘on the fly’ without the need for manual consignment handling and label scanning. As a further step, ‘Thinking Tags’ could be used instead of passive RFID tags. Such tags, which have been developed within the project, combine active short-range communication capabilities with sensing, memory, and computing power. Key issues in their design include low power consumption and low costs.

Thinking Tags will offer opportunities far beyond the mere transmission of static identification information, including, but certainly not limited to:

- continuous measuring and monitoring of environmental conditions (temperature, humidity) for sensitive shipments (e. g., frozen food) at individual item level,
- active alerting of the owner of a shipment in case of an alarm, i. e., deviation from the planned transport route, inadequate environment conditions, etc.,
- recording of the history (location, environmental conditions, status) of a shipment in order to provide evidence in case of liability issues.

5 Conclusions

In this paper we presented the ParcelCall approach towards an open architecture for tracking and tracing in transport and logistics.

The de-centralised architecture has several attractive features with respect to the identified requirements. Most importantly, this architecture scales extremely well; it is no problem to install an additional server if need be. Almost as important, there is no need to modify existing corporate IT infrastructures. The only thing that needs to be done is specify and implement an interface between the infrastructure and the GTS. If required, incoming information (from the MLS) can first be processed internally before it is made available to the public via the GTS network (for instance, if exact location information must not be made available for security reasons). Moreover, small companies can compete on a more level playing field.

Internal details, such as change of transport mode or use of a sub-contractor are hidden from the end-user, to whom a virtual global delivery system is presented. (Mobile) end-users (i.e., consignors and consignees) can obtain information about a consignment from the Goods Information Server (GIS). The GIS holds the individual user profiles, checks and verifies a user’s identity, forwards the query to an appropriate GTS and returns the response to the user’s current end system.

We believe that customers will benefit from improved information on their shipments; their potential benefits include, but are not limited to improved planning, and better management of supply chains and inventories.

Likewise, the European transport and logistics industry will greatly benefit from a unified architecture for the exchange of continuous tracing information. It will enable the deployment of new products and services and the improvement of existing ones.

Acknowledgement

Part of the work described in this paper was funded by the European Commission; project # IST-1999-10700.

References

- [1] Busboom, A.; Jakobs, K.: Intermodal End-to-End Tracking and Tracing - Introducing the ParcelCall Approach. Proc. 7th World Congress on Intelligent Transport Systems, 2000.
- [2] Spinardi, G.; Williams, R.; Jakobs, K.; Pils, C. (eds): System requirements and initial system concept, ParcelCall IST-1999-10700, Deliverable D01, 2000. <http://www.ParcelCall.com/publications/d01-otf.pdf>
- [3] Jakobs, K.: CEN/ISSS Workshop on Multi-modal End-to-End Tracking & Tracing (WS-MEET) – Business Plan. <http://www.nen.nl/wsmeet/download/meet025.pdf>
- [4] Jakobs, K.; Graham, I.; Lloyd, A.; Spinardi, G.; Williams, R.: An Integrated Approach Towards Next Generation Tracking & Tracing. Proc. 10th International Conference on Management of Technology, IAMOT, 2001.

Distributed Management of High-Layer Protocols and Network Services through a Programmable Agent-Based Architecture

Luciano Paschoal Gaspar, Luis Felipe Balbinot, Roberto Storch,
Fabricio Wendt, and Liane Rockenbach Tarouco

Federal University of Rio Grande do Sul, Institute of Informatics,
Av. Bento Gonçalves, 9500 – Agronomia – CEP 91591-970 – Porto Alegre, Brazil,
paschoal@inf.ufrgs.br, <http://www.inf.ufrgs.br/~paschoal>

Abstract. This paper proposes an architecture for distributed management of upper layer protocols and network services called **Trace**. Based on the IETF Script MIB, the architecture provides mechanisms for the delegation of management tasks to mid-level managers, which interact with monitoring and action agents to have them executed. The paper introduces PTSL (*Protocol Trace Specification Language*), a graphical/textual language created to allow network managers to specify protocol traces. The specifications are used by mid-level managers to program the monitoring agents. Once programmed, these agents start to monitor the occurrence of the traces. The information obtained is analyzed by the mid-level managers, which may ask action agents for the execution of procedures (Perl scripts), making the automation of several management tasks possible.

1 Introduction

The use of computer networks to support a growing number of businesses and critical applications has stimulated the search for new management solutions that maintain not only the physical infrastructure, but also the protocols and services that flow over it. The popularization of electronic commerce (e-commerce) and the increasing use of this business modality by companies, for instance, imply using the network to exchange critical data from the organization and from its customers. Protocols and services that support these applications are critical and, therefore, need to be carefully monitored and managed.

Not only critical applications require special attention. New protocols are frequently released to the market to support an increasing set of specific functionalities. These protocols are quickly adopted by network users. As a result of this fast proliferation, weakly-tested and even faulty protocols are disseminated to the network consuming community. In several cases these anomalies, as well as the miscalculated use of resources, are the cause of network performance degradation and end up unnoticed.

We believe that most of the research carried on so far try to provide mechanisms to guarantee higher availability and performance for networks (e.g. Hood and Ji work on proactive fault detection [2]). While solutions to manage physical network infrastructure are established and tested, it is still needed to investigate ways to provide effective management of applications and protocols.

Existing management tools are not completely prepared to allow the monitoring of these new applications and protocols. Most of the tools only allow the monitoring of a closed set of them. The ability to observe new ones depends on the firmware update of the monitoring hardware (e.g. RMON2 probes [3]) or on the programming in low level languages as the extensible probe architecture proposed by Malan and Jahanian [4]. Due to the complexity of the task, most network managers neglect this possibility.

In some approaches it is possible to recognize and count packet flows specified by simple filtering rules (e.g. `tcpdump`-like filters used by `ntop` [5]) or by descriptive languages such as SRL [6], used by `NeTraMet` [7]. However, these filtering languages lack constructors that allow a rule to be defined as a sequence of packets each with specific filtering options, making it impossible to accomplish time-based or correlated analysis of flows.

Other solutions such as Tivoli Enterprise [8] are intrusive, since they require that developers insert specific monitoring procedure calls while developing applications. This approach is only suitable for applications developed in-house. It cannot be used to manage proprietary protocols (e.g. web browsers and servers and e-mail client and servers). Besides that, one must invest on personnel training to use the monitoring APIs.

Regarding the type of information gathered by monitoring engines, some approaches such as the IETF RMON2 MIB (Remote Network Monitoring Management Information Base version 2) store, for a pre-defined set of high-layer protocols supported by the probe, the number of packets sent/received by a host or exchanged by host pairs. Gaspary et al. describe in [9,10] the advantages and limitations of the RMON2 MIB. One of the RMON2 weaknesses is that it does not store any information related to performance, but it has been discussed by the Remote Network Monitoring group at the IETF [11].

Finally, we should point out that many management tools are limited to monitoring [5,7] and the network manager has to take actions manually when unexpected behaviors from these protocols are observed.

In this paper we present **Trace**, an architecture for distributed management of enterprise networked applications, high-layer protocols and network services [12] based on the IETF Script MIB [13]. Through a graphical and textual language based on finite state machines, the network manager defines protocol traces to be observed. These specifications are readily received by one or more programmable agents that immediately start to check whether a defined trace occurs or not. The observation of these traces in the network traffic triggers actions, which are also determined by the network manager.

The paper is organized as follows: section 2 describes the language to specify protocol traces. In section 3 the architecture is presented. Section 4 illustrates

how to accomplish fault management using the architecture. In section 5 we present a summary and concluding remarks.

2 Protocol Trace Representation Using PTSL

In this section we propose PTSL (*Protocol Trace Specification Language*), a graphical and textual language for the representation of high-layer protocol traces. The languages are not equal. The textual one makes the complete representation of a trace possible, including the specification of both the state machine and the events that trigger the transitions. On the other hand, by using the graphical language one can graphically represent the state machine but only label the events that trigger the transitions.

2.1 Organization of a Specification

The textual specification of a trace begins with the keyword **Trace** and ends with **EndTrace**. Initially, the manager may describe some optional items to the specification (see figure 1, lines 2–7). Next, it is broken down into three sections: **MessagesSection** (lines 8–10), **GroupsSection** (lines 11–13) and **StatesSection** (lines 14–16), where messages to be observed, grouping and state machines that describe the trace are respectively specified.

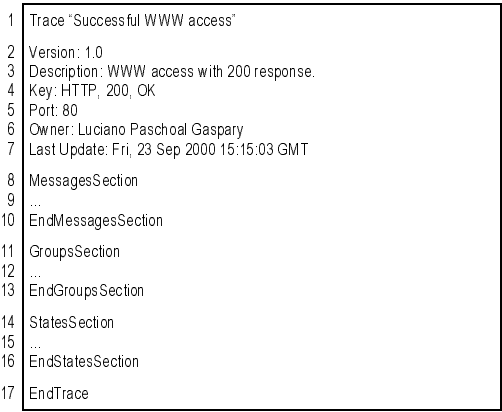


Fig. 1. Schematic representation of a textual specification.

If the trace to be monitored belongs to a single application-layer protocol then the network manager may specify the TCP or UDP port number using the **Port** parameter (line 5). It will simplify packet classification during the monitoring phase.

2.2 State Machines

The trace of a protocol is defined through a finite state machine. The network manager may define a model to monitor just a part of or the whole protocol, or interactions that comprehend more than one protocol. Figure 2 shows two trace examples. In the first example (a), the manager is interested in monitoring the successful accesses to a WWW server. The trace shown in (b) does not describe a single protocol; it is rather made up of a name resolution request (DNS protocol), followed by an ICMP **Port Unreachable** message. This trace occurs when the host where the service resides is on, but the *named* daemon is not running.

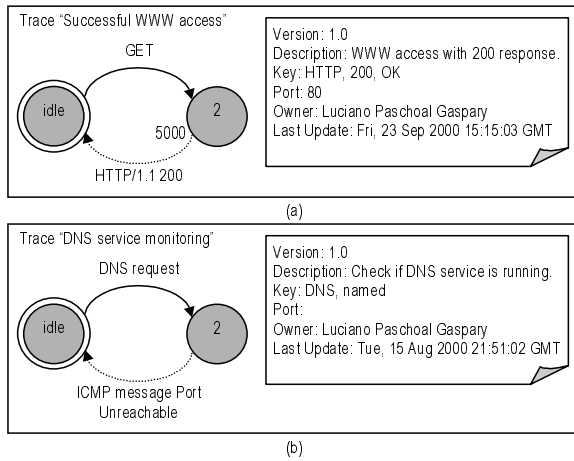


Fig. 2. Graphical representation of a trace. (a) Successful WWW request. (b) DNS request not replied because *named* daemon is not executing.

As one can see states are represented by circles. The initial state has the label *idle* associated to it. The final state is represented by two concentric circles. In both examples the initial and final states are the same (*idle*). Transitions are represented by unidirectional arrows. A continuous arrow indicates that the transition is triggered by the client host, whereas a dotted arrow denotes that it is caused by the server host. The text associated to a transition only labels the event (specified as a message or grouping in the textual language) that will trigger it. It means that the whole specification of a transition only can be done using the textual language. The graphical representation of the state machines shown in figure 2 can be mapped to the textual specification presented in figure 3.

2.3 Transitions

In addition to making a high-level representation of traces, it is necessary to describe what causes the change of states. Before describing the adopted solution, it is important to highlight that high-layer protocols are specified in

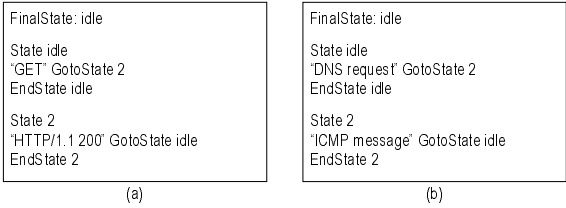


Fig. 3. Textual representation of state machines.

many different ways. Larmouth classifies them as character or binary-based [14]. Character-based protocols are defined as a set of text lines coded in ASCII (e.g. HTTP and SMTP). Binary protocols, on the other hand, are defined as strings of octets or bits (e.g. TCP).

Considering the differences between both protocol types, we propose state transitions to be represented by a positional approach. Taking the example shown in figure 2a, we present (see figure 4a) how to represent the transition HTTP/1.1 200.

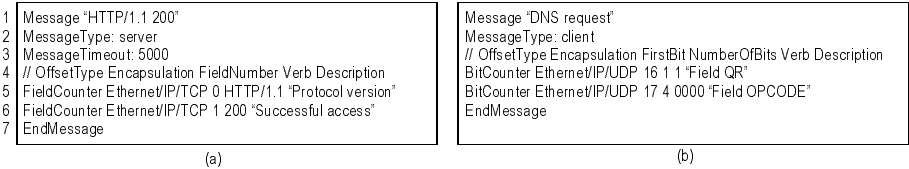


Fig. 4. Representation of (a) Character-based and (b) Binary protocol fields.

As the transition is expected to be triggered by the server host, one must set the **MessageType** field to **server** (line 2). Since both protocol fields (HTTP/1.1 and 200) belong to a character-based protocol, the search for their positions within the packets is made by fields (**FieldCounter**, lines 5–6). In this example, HTTP/1.1 is the first string that appears on the message and therefore its offset is 0 (third parameter in line 5). The second string to appear is 200 and its offset is 1 (line 6). For each protocol field defined in a message it is also necessary to inform where to look for it (encapsulation **Ethernet/IP/TCP**, lines 5–6).

When the transition is caused by a binary protocol, the offset is presented in bits (**BitCounter**). In this case, it is necessary to inform where the field starts (**FirstBit**) and the number of bits to be observed from this offset on (**NumberOfBits**). A standard DNS request can be recognized by two fields: **QR** (when set to 1 indicates a request to the server) and **OPCODE** (when set to 0 represents a standard query). Field **QR** is 16 bits away from the beginning of the header and its size is 1 bit. Field **OPCODE** starts in the seventeenth bit and

occupies 4 bits. In figure 4b the textual representation of a standard DNS request is shown.

It is possible to group one or more messages into one single transition. For example, in figure 2a it would be possible to replace the HTTP/1.1 200 with the grouping HTTP/1.1 2XX. In this case the trace would monitor the rate of all successful WWW operations generated by client requests (2XX) instead of only observing the occurrence of WWW accesses whose return code is 200 (successful request). Figure 5 shows the representation of this grouping (lines 16–18).

```

1 MessagesSection
2 ...
3 Message 'HTTP/1.1 200'
4 MessageType: server
5 FieldCounter Ethernet/IP/TCP 0 HTTP/1.1 'Protocol Version'
6 FieldCounter Ethernet/IP/TCP 1 200 'Successful request'
7 EndMessage
8 Message 'HTTP/1.1 202'
9 MessageType: server
10 FieldCounter Ethernet/IP/TCP 0 HTTP/1.1 'Protocol Version'
11 FieldCounter Ethernet/IP/TCP 1 202 'Request accepted but not processed'
12 EndMessage
13 ...
14 EndMessagesSection
15 GroupsSection
16 Group 'HTTP/1.1 2XX'
17 Messages: 'HTTP/1.1 200', 'HTTP/1.1 202', ...
18 EndGroup
19 EndGroupsSection
20 StatesSection
21 FinalState: idle
22 State idle
23 'GET' GotoState 2
24 EndState idle
25 State 2
26 'HTTP/1.1 2XX' GotoState idle
27 EndState 2
28 EndStatesSection

```

Fig. 5. Representation of message grouping.

In some cases the network manager may be interested in observing the occurrence of a certain string within the data field of a certain protocol, no matter where it is located. To do that, in the definition of such a message one must use `NoOffset` as the `OffsetType` parameter. This feature is interesting, for instance, to observe the attempt of an intrusion. The example presented in figure 6 defines that every TCP packet must be tested for the occurrence of the string `/etc/passwd` (line 4).

We have also created a mechanism to allow the determination of a timeout to a transition to occur. To do that one must associate a timeout value (in

```
1 Message "/etc/passwd"
2 MessageType: client
3 // OffsetType Encapsulation Verb
4 NoOffset Ethernet/IP/TCP /etc/passwd
5 EndMessage
```

Fig. 6. Non-specified offset message field.

milliseconds) to the message definition (see figure 4a, line 3). When not defined, a default value is used by the network monitor.

3 The Trace Architecture

The architecture we propose is an extension of the existing distributed management infrastructure standardized by the IETF [15] with high-layer protocol and network service management capabilities. Figure 7 shows the main components of the architecture. It is composed of management stations, mid-level managers, programmable monitoring agents and programmable action agents. The following sub-sections describe the components of the architecture and their interactions with each other.

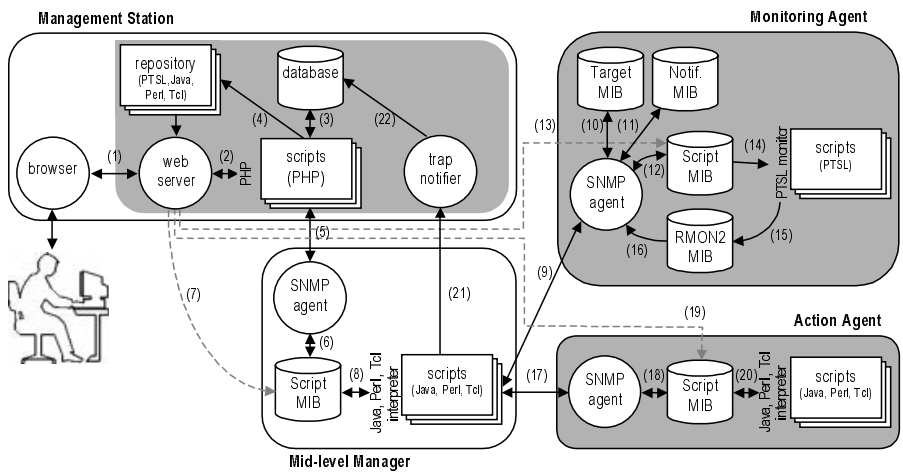


Fig. 7. Components of the architecture.

3.1 Management Station

The most important activities accomplished by the network manager from a management station are (a) registration of mid-level managers, monitoring and

action agents, (b) specification of protocol traces and actions, (c) specification, delegation, observation and interruption of management tasks and (d) receipt and visualization of traps.

As the whole architecture is based on the Script MIB, protocol traces, actions and management tasks are scripts executed by monitoring agents, action agents and mid-level managers, respectively. Protocol traces are specified by the network manager using the PTSL language. Actions are scripts developed using Java or any scripting language such as Tcl and Perl. Management tasks may also be implemented using any language and coordinate monitoring and action agents. Such a script programs the monitoring agents, observes the occurrence of the trace and activates action agents when a condition associated to a protocol trace holds. The same script may also report events to the management station raising traps.

At the management station the network manager can specify traces using a graphical tool (see an example of such a tool in figure 8) or, if he knows the language, by editing a text file. The same occurs with actions and management tasks. The specification of protocol traces, actions and management tasks are stored in the database (figure 7, see flows (1, 2, 3) in diagram). When a management task is about to be delegated, they are mapped to files and stored in the repository (4).

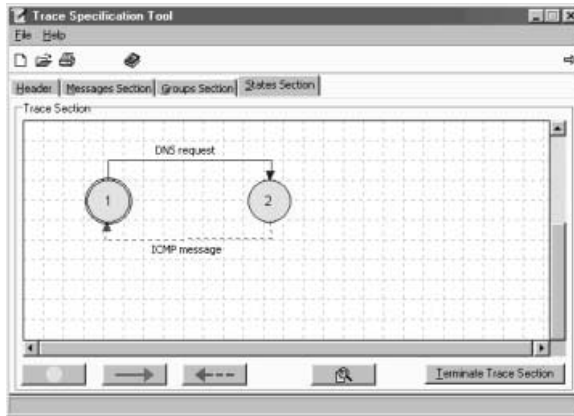


Fig. 8. Prototype of the tool for trace specification.

Communication between the management station and the mid-level managers takes place using the SNMP protocol (Script MIB) (5, 6). The manager can delegate a management task to a mid-level manager as well as abort it at any time. Intermediate and final results of the execution of a management task are stored directly at the Script MIB of the mid-level manager responsible for the task and can be retrieved by the management station using the SNMP protocol (5, 6).

The manager may receive traps through an element called *trap notifier* (21). When received, all traps are stored in a database (22). The traps are permanently retrieved by a script (3) that updates the manager's web browser (2, 1) using the push technology.

3.2 Mid-level Manager

Mid-level managers execute and monitor management tasks delegated by the management station and report the most important events to it. The number of mid-level managers is determined by the network manager and depends on the size and complexity of the infrastructure to be managed.

The process of configuring mid-level managers is the following: the network manager defines a management task and stores it at the repository (1, 2, 3, 4). Next, the activation of the task must be scheduled using the Script MIB (5, 6). In order to do that, the mid-level manager has to be informed about the location of the task (script). When activated, the task is retrieved from the repository using the HTTP protocol (7) and executed (8).

The script executed by the mid-level manager installs the protocol trace (9, 12) and the action script (17, 18), requests the monitoring agent to start observing the occurrence of the protocol trace just installed (9, 12), polls RMON2 variables periodically to monitor the occurrence of the trace (9, 16) and, depending on what is observed, dispatches the execution of the action script (17, 18) or raises a trap to the manager (21). The script communicates with the agents using the SNMP protocol.

The same script may also subscribe at monitoring and action agents to the traps it wants to receive. The Target MIB is used to identify the management task (IP address and UDP port) (9, 10). Using the Notification MIB the mid-level manager indicates (through filters) which traps should be sent to the management task, whose location was identified at the Target MIB (9, 11) [16]. When the script receives a trap, it may dispatch the execution of an action (17, 18) or correlate it with previously received traps. The result of these operations may be informed to the management station (21).

3.3 Monitoring Agent

The monitoring agents are responsible for observing the traffic on the network segment where they are installed. They are configured by mid-level managers and are called programmable because they are able to monitor protocol traces delegated dynamically by the network manager. This flexibility is obtained through the language presented in section 2. When the mid-level manager sets the monitoring agent up (9, 12), the former defines which protocol trace it should retrieve (it is indicated within the script that implements the task). Once retrieved (13), the trace file is loaded by the monitoring engine and the observation starts (14).

Whenever the occurrence of the trace is observed between any pair of hosts, information is stored within an RMON2-like MIB (15). This MIB is different from the standard because the `protocolDir` group is writable in our approach.

Therefore the probe stores statistics according to the protocol traces of interest to the network manager. Additionally, the granularity of the monitoring becomes higher. Instead of storing overall statistics on traffic generated by a given protocol, statistics are generated according to the occurrence of specified traces or transactions.

The **alMatrix** group from RMON2 MIB stores statistics on the trace when the latter is observed between every pair of hosts. Table 1 shows the contents of the **alMatrixSD** table. It gathers the observed number of packets and octets exchanged between every pair of hosts (client/server) using the protocol traces being monitored by the probe. In the example, two traces were observed: **Successful WWW access** and **DNS service monitoring** (previously shown in figure 2a and b).

Table 1. Information obtained by referring to the **alMatrixSD** table.

Source Address	Destination Address	Protocol	Packets	Octets
17.16.10.1	17.16.10.2	Successful WWW access	254	120.212
17.16.10.6	20.24.20.2	Successful WWW access	20	10.543
17.16.10.1	17.16.10.33	DNS service monitoring	4	4.350
17.16.10.32	17.16.10.33	DNS service monitoring	8	7.300

The disadvantage of using RMON2 MIB is that it does not have objects capable of storing information related to performance. For this reason, our group is currently considering the possibility of using, in addition to that, an RMON2 extension, such as Application Performance Measurement MIB [11]. Table 2 shows the kind of information stored by this MIB. The first line shows that the trace **Successful WWW access** has been observed 127 times between hosts 17.16.10.12 and 17.16.10.2. Additionally, the mean response time was 6 seconds.

Table 2. MIB with information on performance.

Client	Server	Protocol	Success.	Unsuccess.	Responsiv.
17.16.10.12	17.16.10.2	Successful WWW access	127	232	6 sec.
17.16.10.12	20.24.20.2	Successful WWW access	232	112	17 sec.
17.16.10.1	17.16.10.33	DNS service monitoring	2	0	3 sec.

3.4 Action Agent

Action agents reside in hosts where network services are executed. Their function is to perform a given operation on these services. Let us take as example the DNS service. Figure 2b shows a trace that enables to detect when the *named*

daemon is not in execution. There may be a management task, delegated to the mid-level manager, that monitors the occurrence of this trace. If that is the case, the action to be taken is to contact the action agent (see flow (17) in figure 7), which is located in the host where the DNS service is installed, and request the execution of a script to restart the daemon (18, 19, 20) (see the example of such a script developed in Perl in figure 9). The result obtained by the action agent is accessible to the mid-level manager (17, 18), which may send it to the manager for notification purposes (21).

```
#!/usr/bin/perl

my $pid;

# Verify if the process named is executing.
if (-e "/var/run/named.pid") {
    $pid = "/bin/cat /var/run/named.pid";
}

# If named is running, restart it using a HUP signal, otherwise
# instantiate the process again.
if (defined $pid) {
    print "Restarting named (sending HUP signal)...\\n";
    "/bin/kill -HUP $pid";
} else {
    print "Starting named (was not running)...\\n";
    "/usr/sbin/named &";
}

# Test if the process is executing.
if (-e "/var/run/named.pid") {
    $pid = "/bin/cat /var/run/named.pid";
    print "The named daemon is up and running as PID $pid\\n";
} else {
    print "The named daemon could not be started!\\n";
}
```

Fig. 9. Perl script to restart the *named* daemon.

4 Fault Management of Network Services

Fault management is an important target of the proposed architecture. An example of fault management concerning to high-layer protocols and network services is checking the availability of a network service and restart it if it is not running. Figure 10 shows how this can be achieved using the architecture. In this case the task delegated to the mid-level manager is supposed to monitor the DNS service.

This monitoring is performed by sniffing the packets seen in the segment. In a situation where the daemon responsible for the DNS service is not running, the agent will observe a DNS request and some time later a **Port Unreachable** ICMP message from the serving host. In this case the mid-level manager should contact an action agent, which resides in the DNS serving host and request the execution of a script to restart the daemon (such as the one illustrated in

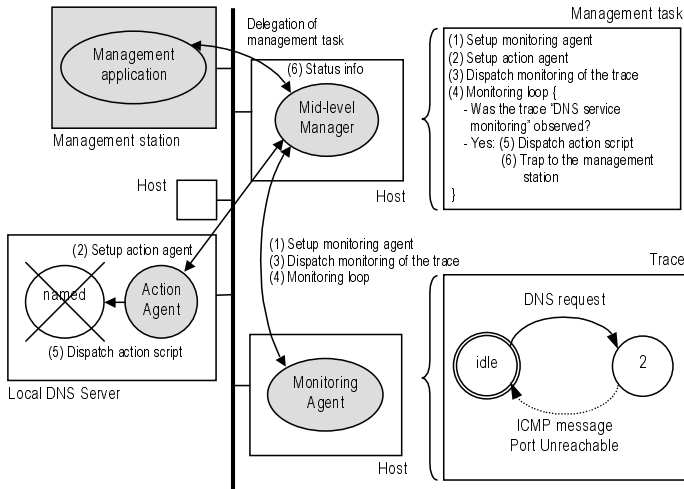


Fig. 10. Execution steps of a management task.

figure 9). The textual specification of the trace **DNS service monitoring** is shown in figure 11.

The architecture was designed to take into account all the standard functional areas of management: fault, configuration, accounting, performance and security (FCAPS). Our research group has explored in [12] its characteristics to validate the usefulness of the architecture for the management of high-layer protocols and network services.

5 Conclusions

This work presented a distributed architecture for the management of high-layer protocols and network services based on programmable agents. Motivated by the growing need companies have to monitor high-layer protocols and their critical applications, the work proposes a flexible architecture able to follow the fast dissemination of protocols and networked applications (that need to be managed). The architecture may be used either in corporate networks or in application service providers.

The most important contribution of the architecture is the granularity of the monitoring. The observation of network traffic on a transaction basis makes the understanding of protocol and networked application behaviors possible. The language proposed to specify protocol traces is simple, but the network manager has to know the format of the packets exchanged by the application or protocol to be managed.

Another significant contribution of the architecture is the possibility to do more than just monitoring. Management tasks provide the manager with mechanisms to monitor the occurrence of protocol traces and to dispatch management

```

Trace "DNS service monitoring"

Version: 1.0
Description: Trace to detect when named is not running
Key: named, fault, DNS service
Port:
Owner: Luciano Paschoal Gaspary
Last Update: Tue, 10 Aug 2000 15:30:58 GMT

MessagesSection

Message "DNS Request"
// See code in figure 4b.
EndMessage

Message "ICMP Message"
MessageType: server
// OffsetType Encapsulation FirstBit NumberOfBits Verb Description
BitCounter Ethernet/IP 0 8 00000011 "Type field=00000011?"
BitCounter Ethernet/IP 8 8 00000011 "Code field=00000011?"
EndMessage

EndMessagesSection

StatesSection

// See code in figure 3b.

EndStatesSection

EndTrace

```

Fig. 11. Trace to monitor the DNS service.

scripts executed by programmable action agents. These mechanisms contribute to management automation in some scenarios.

As described by Strauß[17], “distributed management in general [18] and the Script MIB specifically are expected to bring various advantages over the centralized concept suited for the raising demands in network management. A commonly mentioned advantage is the increased scalability due to the delegation of management tasks from the centralized network management station to mid-level managers. This implies that CPU and network load is also delegated to the subnets to which the mid-level managers belong. Another major advantage is concerned with the robustness of management tasks. While centralized management systems require a reliable network, the distributed approach allows to delegate some sensible manager functions next to the observed agents. Hence these functions may become independent from less reliable WAN links, for example”.

The architecture requires more work to be controlled than a single centralized management system. The management of its components becomes more complicated. It is necessary to distribute and update scripts, control running scripts, gather and correlate intermediate and final results. We believe that such operations, as well as the specification of traces, can be simplified by adding an easy-to-use interface to the management application. Currently, our research group is working on the improvement of the prototype. After that, a larger scale validation will be done.

References

1. L. L. Ho, C. J. Macey, and R. Hiller. "A Distributed and Reliable Platform for Adaptive Anomaly Detection in IP Networks". In *Proc. 10th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, Zurich, October 1999, pp. 33–46.
2. C. S. Hood and C. Ji. "Intelligent Agents for Proactive Fault Detection". *IEEE Internet Computing*, vol. 2, no. 2, pp. 65–72, March/April 1998.
3. S. Waldbusser. "Remote Network Monitoring Management Information Base Version 2 using SMIv2". Request for Comments 2021, January 1997.
4. G. Malan and F. Jahanian. "An Extensible Probe Architecture for Network Protocol Performance Measurement". In *Proc. SIGCOMM*, Vancouver, September 1998.
5. L. Deri and S. Suin. "Ntop: Beyond Ping and Traceroute". In *Proc. of the 10th IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, Zurich, October 1999, p. 271–283.
6. N. Brownlee. "SRL: A Language for Describing Traffic Flows and Specifying Actions for Flow Groups". Request for Comments 2723, October 1999.
7. NeTraMet. Available at <http://www.auckland.ac.nz/net/Internet/rtfm/>.
8. C. Cook et al. "An Introduction to Tivoli Enterprise". First edition. USA: International Technical Support Organization, 1999. Available at <http://www.redbooks.ibm.com>.
9. L. P. Gasparly and L. R. Tarouco. "Characterization and Measurements of Enterprise Network Traffic with RMON2". In *Proc. 10th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, Zurich, October 1999, pp. 229–242.
10. L. P. Gasparly and L. R. Tarouco. "Managing Users, Applications and Resources with RMON2". In *Proc. Global Telecommunications Conference, Symposium on Enterprise Applications and Services*, Rio de Janeiro, December 1999, pp. 1997–2001.
11. S. Waldbusser. "Application Performance Measurement MIB". Internet Draft, November 2000.
12. L. P. Gasparly, L. F. Balbinot, R. Storch, F. Wendt, and L. R. Tarouco. "Towards a Programmable Agent-based Architecture for Enterprise Application and Service Management". In *Proc. First IEEE/IEC Enterprise Networking Applications and Services Conference*, Atlanta, June 2001.
13. D. Levi and J. Schönwälder. "Definitions of Managed Objects for the Delegation of Management Scripts". Internet Draft, Nortel Networks, TU Braunschweig, July 2000.
14. J. Larmouth. *ASN.1 Complete*. Open Systems Solutions, 1999.
15. J. Schönwälder, J. Quittek, and C. Kappler. "Building Distributed Management Applications with the IETF Script MIB". *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 5, pp. 702–714, 2000.
16. D. Levi, P. Meyer, and B. Stewart. "SNMP Applications". Request for Comments 2573, April 1999.
17. F. Strauß. "Advantages and Disadvantages of the Script MIB Infrastructure". Project Report, October 2000. Available at <http://www.ibr.cs.tu-bs.de/projects/jasmin/>.
18. J. Schönwälder. "Network Management by Delegation – From Research Prototypes Towards Standards". *Computer Networks and ISDN Systems*, vol. 29, no. 15, pp. 1843–1852, November 1997.

A Buffer-Management Scheme for Bandwidth and Delay Differentiation Using a Virtual Scheduler

Roman Pletka¹, Patrick Droz¹, and Burkhard Stiller²

¹ IBM Research, Zurich Research Laboratory,
8803 Rüschlikon, Switzerland
`{rap,droz}@zurich.ibm.com`

² Computer Engineering and Networks Laboratory (TIK), ETH Zürich,
8092 Zürich, Switzerland
`stiller@tik.ee.ethz.ch`

Abstract. This paper presents a new scalable buffer-management scheme for IP Differentiated Services. The scheme consists of a Differentiated Random Drop (DRD) algorithm using feedback from a virtual scheduler. DRD choses a queue to perform an early packet drop to avoid congestion according to a specific probability function. It will be shown that DRD in conjunction with first-come first-served scheduling is able to support relative service differentiation. The virtual scheduler is introduced to enable service differentiation in terms of bandwidth and delay at the same time. A virtual scheduler runs in parallel to the real scheduler and maintains virtual queue lengths that are being used by the congestion avoidance scheme for packet-drop decisions. Scheduling packets for transmission is performed by the real scheduler only.

1 Introduction

In the past few years many different scheduler and queue-management algorithms have been proposed. Research activities have been and still are focused on how to satisfy the Quality-of-Service (QoS) requirements of higher-priority flows while keeping fairness among classes and preventing starvation of low-priority traffic.

In Weighted Fair Queueing (WFQ) schedulers such as Self-Clocked Fair Queueing (SCFQ) [1] and other rate-proportional service disciplines [2,3,4], queue weights are used to provide per-flow¹ bandwidth guarantees: The link share of backlogged connections is proportional to the queue weight, and excess bandwidth is distributed in the same manner. Flows that are significantly below their reserved bandwidth share will experience less delay. This means that delay-sensitive services such as Voice-over-IP (VoIP) can be implemented using WFQ

¹ In general the term micro-flow is defined by the 5-tuple of source and destination address and port number, and the protocol number in the IP header of the packet. Macro-flows may consist of a large number of micro-flows forming a flow aggregate. For the sake of simplicity, this paper refers to flow aggregates simply as flows, and accordingly to a single micro-flow as a flow.

only in combination with a token bucket at the ingress node and that the token bucket rate is set lower than the reserved rate. In Class-Based Queueing (CBQ) [5], delay differentiation can be achieved using a priority-based packet-scheduling algorithm for bounded traffic classes. In all these schemes, thresholds (for scheduler and token buckets) have to be set carefully, and often their meaning is not intuitive. Usually, this is done in a static manner when the network is set up, and often default parameters are not modified at all.

This paper focuses on a new threshold-based buffer-management scheme that consists of a combination of Differentiated Random Drop (DRD) [6] and virtual scheduling. Unlike other known schemes, the proposed scheme supports simultaneous bandwidth and delay differentiation and has the following advantages:

- Dynamic drop rate adaption of traffic classes,
- efficient and early congestion avoidance, and
- easy setting up of thresholds.

It will be shown that the scheme is suitable for a Diffserv [7] enabled network, where it can be used to implement relative QoS guarantees in Assured Forwarding (AF) [8] per-hop-behaviors.

The elements of active buffer management are algorithmic droppers, packet-marking strategies, and scheduling algorithms. Several extensions that are combined in a flow-and-queue threshold-based buffer-management scheme use active buffer-management elements. These extensions are fundamental to fair packet dropping and better overall buffer usage. Furthermore DRD is briefly introduced as an efficient congestion avoidance algorithm. It will be shown that DRD in conjunction with threshold-based buffer management and simple first-come first-served (FCFS) scheduling is able to provide service differentiation in terms of dynamic and adaptive packet drop rates, which are relative to other queues in the system. The efficiency of a virtual scheduler, which is the key to bandwidth and delay differentiation, is compared to that of a simple FCFS scheduler using extensive simulations realized in a modified version of the network simulator *ns* [9] that was specifically extended for this purpose.

The remainder of this paper is organized as follows. Based on the detailed discussion of the threshold-based buffer management in Section 3, the virtual scheduler is introduced in Section 4. A simulation model utilizes the Diffserv Architecture to provide insights into the combination of FCFS and DRD with a virtual scheduler (Section 5). Finally, Section 6 summarizes the advantages and draws conclusions.

2 Related Work

In [10] Drovolis et al. propose a proportional differentiation model to refine and quantify relative service differentiation. Two packet schedulers that approximate the model are introduced and evaluated in simulations. The proportional model is applied on queueing-delay differentiation only and leaves the problem of coupled delay and loss differentiation for future work.

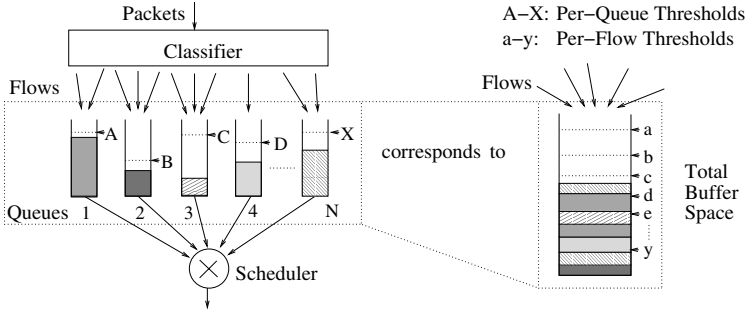


Fig. 1. Architecture of buffer management.

In [11] Risso discusses Decoupled Class Based Scheduling (D-CBQ), a CBQ-derived scheduling algorithm that uses new link-sharing guidelines to decouple bandwidth and delay for bounded classes. The algorithm improves delay characteristics of bounded classes compared to CBQ. Whereas setting higher priorities (that means lower delays) no longer leads to more bandwidth being allocated, incoming traffic still has to be limited by means of an additional token bucket filter. Note that the impact on delay and bandwidth using unbounded classes has not been studied in a severely overcharged network environment.

3 Threshold-Based Buffer Management

Figure 1 shows a flow-and-queue threshold-based buffer-management scheme [6]. Thresholds are assigned to flows and queues. As can be seen on the left-hand side, each of these flows is attributed to one queue and several flows can enter the same queue. In general, packets with the same QoS needs will enter the same queue although there may be multiple queues having approximately the same properties to differentiate for example between TCP and UDP traffic. On the right-hand side, the same flows are shown in the context of overall buffer space. The process of packet classification will not be discussed as it would exceed the scope of this paper.

As indicated by its name, flow-and-queue threshold-based buffer management is a scheme primarily based on two thresholds. The first threshold limits global buffer occupancy of a flow and is called the *per-flow threshold*. This means that flows exceeding their per-flow threshold undergo a special treatment such as marking or dropping packets. Marking and dropping depends on the type of buffer management and will be discussed later. Per-flow thresholds are measured relative to the total buffer space used.

The second threshold is a *per-queue threshold*, which allows a segmentation of the available buffer space and is compared to the buffer space used by this queue. When the per-queue threshold is exceeded, packets have to be dropped to limit the maximum packet delay. When used without additional strategies, the per-queue threshold acts as a “hard” dropping policy. Packets may suddenly be dropped in bursts when the queue size exceeds the threshold. Clearly this behav-

ior is not desirable. An early dropping policy such as Random Early Detection (RED) [12] or DRD should be combined with this threshold. For this purpose DRD will be introduced in the next Section. Later it will be shown that DRD outperforms traditional RED, in addition to having other useful properties.

Using per-queue thresholds allows more than the real existing buffer space to be allocated to queues as opposed to hard segmented buffer spaces as used in [13]. This means that the sum of all per-queue thresholds may exceed the total available buffer space. The advantage of such a strategy is that it supports larger bursts of a flow when other flows are on a low buffer-usage level or not backlogged at all and, therefore, uses less global buffer space. On the other hand when all flows send at a peak rate, the fixed per-queue buffers cannot be fully exploited at the same time. At this point the per-flow threshold will act as a limiter. In conjunction with RED or DRD, this limit is not a hard limit and therefore does not cause bursty packet drops. The service rate will no longer be absolute but rather relative to other classes. This is even mandatory for giving best-effort traffic the capability of taking advantage of unused bandwidth.

3.1 Hard Dropping Scheme

The simplest buffer-management scheme known consists of dropping packets when no more buffer space is available. This strategy, commonly used in the past and even today, turns out to be inadequate for performing efficient and fair packet forwarding even when used with fair queuing. Packets are often dropped in “bursts” from a single flow, whereas other flows increase their traffic even more. As a result, fairness suffers and QoS requirements simply cannot be guaranteed.

Adding flow-and-queue threshold-based buffer management allows buffer sharing and priority handling. In addition to being dropped when no buffer space is available, packets are dropped when one or both thresholds are exceeded. The simulations discussed in [6] show that such a simple flow-and-queue threshold-based buffer management is not sufficient per se to guarantee services as defined in Diffserv.

In general, the thresholds used in this mechanism act as a hard limit. During congestion periods no indication is performed, and packets are suddenly dropped in large bursts once a threshold has been exceeded. There must be some additional packet-dropping strategies to avoid bursty drops and synchronization of TCP sources.

3.2 Softening Hard Limits

One way to overcome the hard dropping nature of a threshold is to introduce a steadily increasing probability function depending on the average queue size. A linearly increasing function is used because of its simplicity: it is sufficient to add an additional threshold, which can be as simple as a default percentage of the per-flow threshold. The two thresholds together are then used as a lower and upper limit (max/min per-queue thresholds).

The size of the linearly increasing section has to be set relative either to the global buffer space or to the allocated queue space. Too small a value does not

overcome the bursty drop problem and too large a size of the linearly increasing section will introduce premature packet drops. Without going into more details concerning the optimum setting, which would be beyond the scope of this paper, experiments have shown that a value of approximately 50% is reasonable [12].

3.3 Introducing Packet Marking

Service differentiation within a flow can be achieved by introducing a set of drop precedences. A simplistic approach would be to mark all packets that have a higher precedence than the low default drop precedence. In doing so, almost exclusively packets having a higher drop precedence will be dropped, and differentiation between more than two drop precedences will no longer be feasible. Therefore, the proposition is to assign per-flow thresholds to drop precedences, and each drop precedence has its own per-flow threshold. Multiple flows with different per-flow thresholds may coexist in the same queue. Marked packets in a queue may belong to different flows, and no distinction according to the initially given drop precedence is done. The marking is used when drop decisions have to be made. The per-flow drop rate increases with decreasing per-flow threshold. Packet marking can be regarded as a “previous conviction” with a scope that is strictly limited to the actual router.

If per-flow thresholds are used to trigger packet drops, a small per-flow threshold will completely prevent a flow from obtaining any service when network traffic is high. A part of the global buffer space remains unused because it is reserved for other traffic classes that perhaps will not occupy this space in the near future. One way to improve buffer usage is to increase the per-flow threshold but then service differentiation becomes more difficult because these thresholds move closer together. Nevertheless, an arriving packet belonging to such a flow could be enqueued and marked. If packets need to be dropped later, those marked should be dropped first. As a result buffer space is used more efficiently, and overall more packets are served. A sophisticated packet-dropping scheme can take into account the per-queue buffer usage as well as a relative queue priority, and then select a packet to be dropped.

3.4 Algorithmic Dropper and Congestion Avoidance Scheme

Upon packet arrival, the algorithmic dropper examines whether the packet should be enqueued and an additional action taken. An additional action is an action that tries to prevent congestion in the network. Figure 2 illustrates the algorithmic dropper that consists of three parts: The first is used for congestion avoidance and evaluates whether an existing packet has to be dropped in one of the queues by choosing a queue randomly. If yes, a packet drop in this queue is triggered. The second part consists of hard dropping limits (tail-drop) for queue and buffer overflows. The congestion avoidance scheme should drop packets earlier so that tail-drops occur only rarely. In the third part, packet marking to implement drop precedences in a queue occurs.

The congestion-avoidance block uses DRD to evaluate potential packet drops. The main goal of the DRD scheme is to introduce a dynamic per-queue drop

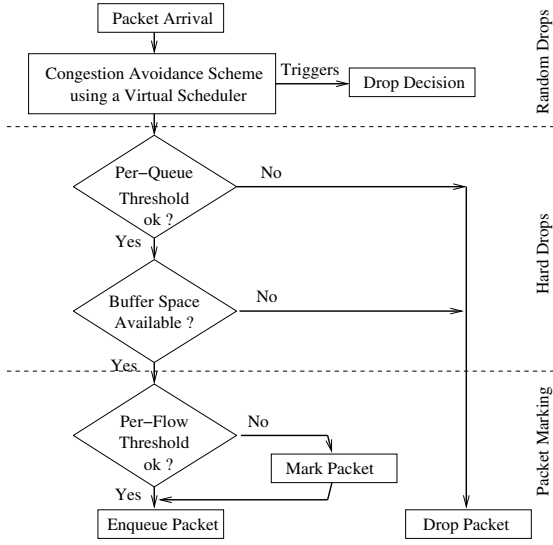


Fig. 2. Flow-chart diagram of an algorithmic dropper.

probability while adding relative dependency among the various queues in the system. This will primarily allow service differentiation such as “better than” another class. Each time a packet arrives, the following congestion-avoidance mechanism is performed before processing of that packet continues. Using a dynamic per-queue probability, one of the queues is chosen randomly and random early discard is then performed in this queue. The per-queue probability p_i is evaluated as follows: Every queue is assigned a fixed priority equal to the queue number i . Thus the queues are sorted according to their priority. The per-queue probability is proportional to the number of bytes in the queue plus the number of bytes in all higher-priority queues. This is a more general approach than is used for RIO in [14]. Clearly queues containing no packets have zero per-queue probability. Note that priorities are introduced only for dropping behavior and not, as for example in CBQ [5], as a per-queue priority used for scheduling purposes. In addition, higher priority does not imply lower packet delay. The per-queue probability p_i can be written as

$$p_i = \begin{cases} C \sum_{k=1}^i b_k & \text{if } b_i \neq 0 \\ 0 & \text{if } b_i = 0 \end{cases}, \quad (1)$$

where b_k is the number of bytes in queue k . For N queues the normalization that leads to the constant C is then given by

$$\sum_{i=1}^N p_i = 1 \Rightarrow C = \frac{1}{\sum_{j=1, b_j \neq 0}^N \sum_{k=1}^j b_k}. \quad (2)$$

Finally, if a packet in that queue has to be dropped, by preference a marked one will be chosen.

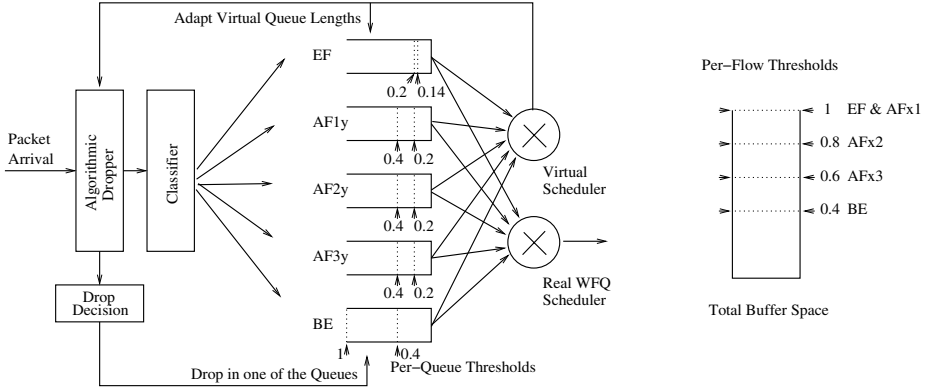


Fig. 3. Architecture of buffer management.

4 Virtual Scheduler

The above-mentioned scheme is able to support relative service differentiation even with a simple FCFS scheduler [6]. Service classes in terms of “better than” can be implemented, and differentiation is expressed in lower drop rates for lower-numbered (higher-priority) queues. Whereas the scheme is able to support minimum bandwidth guarantees and fair excess bandwidth allocation, it fails in differentiating packet delays due to the simple FCFS scheduler. The idea is to combine two schedulers while keeping their advantages: The first scheduler will maintain fair packet scheduling and enable delay differentiation. For this a WFQ scheduler can be used. The second scheduler will be responsible for early congestion avoidance and will start dropping packets if necessary. It maintains virtual queue lengths used by the congestion avoidance scheme as a feedback. This scheduler is called virtual because it does not directly influence the departure time of packets in the buffer. Its result, the virtual queue lengths, are only used by the algorithmic dropper to perform drop decisions.

Figure 3 illustrates the architecture of the buffer-management scheme using a virtual scheduler. As basis the scheme of Section 3 has been taken and enhanced to support a virtual scheduler. The buffer is divided into several queues. The number of queues is configurable, and the queues are served in a WFQ manner. For each queue several parameters are given (queue number, max/min per-queue threshold and queue weight). In contrast to other schemes, these parameters are fixed at the beginning once and for all, and no tuning is required later. In addition, the parameters in the queues with relative delay differentiation are equal (queues 2 to 4), thus making configuration easy. In Section 5 it will be shown that such a scheme is capable of providing a dedicated service class to a given queue. When a packet arrives, it will go through the algorithmic dropper with the only modification that virtual queue lengths are used rather than the real ones to trigger a packet drop. Meanwhile all packets are also served by the virtual FCFS scheduler.

Buffer management becomes quite difficult because the two schedulers serve packets simultaneously. Therefore, a special packet tag that is attributed to each of the packets and contains all necessary information has been introduced. In other words, if a packet has been treated by the real scheduler (and therefore has been sent to the outgoing link) but not yet being served by the virtual scheduler, the packet tag will remain stored in memory while the space used for the real packet can be freed. If a packet has been treated first by the virtual scheduler but not yet by the real scheduler, the packet tag and the packet itself will remain stored in memory until the packet has been served by the real scheduler. It is clear that by introducing packet tags, which can remain in memory longer than a packet's lifetime, overall memory usage will increase. Section 5 discusses this issue, and shows that the increase in memory is limited.

4.1 Parameter-Setting Guidelines

The following guidelines should help set the parameters of a flow-and-queue threshold-based buffer-management scheme with N queues:

- The queue weight w_i corresponds to the minimum bandwidth guarantee for the service class in queue i and is a part of Service Level Agreements (SLA).
- For equal per-queue threshold settings, lower-delay classes are in higher-numbered queues.
- Delay-insensitive queues get a high per-queue threshold to profit from unused buffer space.
- For classes that contain adaptive flows, the minimum per-queue threshold is set to half the maximum per-queue threshold. Non-adaptive flows do not react to packet drops and have equal min/max per-queue thresholds.
- Drop precedences can be set equal over a set of queues. The higher the per-flow threshold, the lower the drop probability. Setting the per-flow threshold to 1 disables packet marking for this flow, but packets can still be dropped in a severely overloaded network.

5 Simulation Results

The simulations described here have been made in a Diffserv-enabled network environment. Flow-and-queue threshold-based buffer management maps well to the Diffserv classes [8,15] for the following reasons:

Number of queues: For a system to scale well, the number of queues is important. It is clear that the flow-and-queue threshold-based buffer management does not treat micro-flows individually, but only applies the service defined for the corresponding service class. Thus, the buffer-management system keeps the queue number low, in a Diffserv environment generally not more than several dozens. Packet classification is based on the Diffserv Codepoint, but other classification rules could also be envisaged.

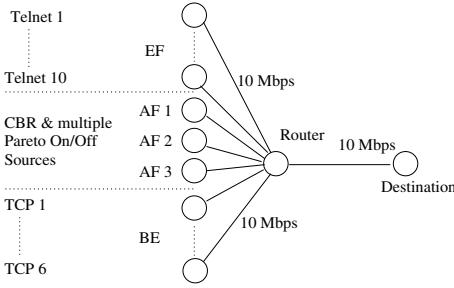


Fig. 4. Simulation topology.

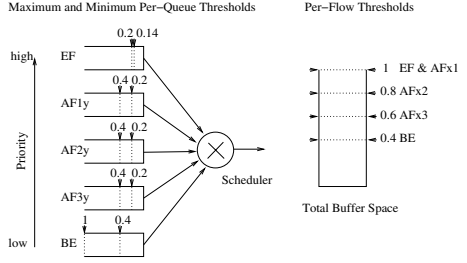


Fig. 5. Per-flow and per-queue thresholds.

Packet delay: Setting a low per-queue threshold assures low packet delay, and bandwidth is guaranteed by the queue weight of the WFQ scheduler. This can be used to implement Expedited Forwarding (EF) per-hop-behavior.

Relative service classes: Using a virtual scheduler, service classes in terms of “better than” or Olympic service [8], which consists of three service classes, namely gold, silver and bronze, can be implemented. The Diffserv AF per-hop-behavior can be used to identify the service class of a packet.

Support of drop precedences: Various packet markers have been proposed in the Diffserv working group [16,17]. These markers use the result of a traffic meter to set the appropriate Diffserv Codepoint (DSCP). They should not be confused with packet marking as introduced in this paper. The marking strategy proposed here differs from these Diffserv markers because it acts only locally in a router. Per-flow thresholds are assigned to Diffserv drop precedences in AF to fulfill dropping differentiation. Although packets are only either marked or not, this is sufficient to support multiple levels of drop precedences. The DSCP is not modified in the process, but can influence the marking done by the scheme.

5.1 Service Differentiation for AF without Virtual Scheduler

In this Section FCFS and SCFQ schedulers without a virtual scheduler are compared. These two scheduler types have been chosen to discuss their main properties when combined with DRD and to show that these properties cannot be maintained at the same time. The topology of the simulation is shown in Figure 4, where multiple sources as given in Table 1 share the same outgoing link at a router. The router uses the DRD scheme as explained in Section 3.4. The first queue is assigned to an EF Diffserv class. Ten Telnet applications generate the traffic for this flow. This traffic is substantially lower than the reserved rate, and other flows may borrow from this unused bandwidth. The following three queues treat three AF Diffserv classes: AF1, AF2 and AF3. These flows are generated by CBR and multiple Pareto on/off sources, which create an equal number of all three drop precedences in each AF class. The sources have been chosen such as to be extremely bursty. The average sending rates for all three AF Diffserv sources are equal and vary from 50 to 150% of the allocated bandwidth. The

highest-numbered queue is designated for adaptive best-effort (BE) traffic. A set of greedy TCP connections generates this traffic. All queues have equal weights and, therefore, equal reserved bandwidth. All links are set to 10 Mbit/s. The maximum buffer space is set to 160 kBytes. During the simulation, all sources are sending data at the rates given in Table 1.

Table 1. Traffic sources.

<i>Flow</i>	<i>Sources</i>	<i>Rate [% of reserved rate]</i>
EF	Telnet Sources	
AF1y	CBR & Pareto On/Off	from 50% to 150%
AF2y	CBR & Pareto On/Off	from 50% to 150%
AF3y	CBR & Pareto On/Off	from 50% to 150%
BE	greedy TCP sources	

The buffer settings are shown in Table 2 and Figure 5. The per-queue thresholds are set to guarantee a maximum delay for each class. The terrassing of the per-flow thresholds in an AF class is important to realize drop precedences. The thresholds AFx1, AFx2 and AFx3 are the same for all AF queues. When setting up the thresholds, no differentiation among the same drop precedence of different AF classes has to be performed. All AF classes start dropping packets at the same per-queue limit. The best-effort RED threshold is set to 40% of its per-queue threshold to enable early congestion avoidance even when the buffer space has been almost completely filled up by other sources.

Table 2. Buffer thresholds.

<i>Thresholds</i>	<i>EF</i>	<i>AFx1</i>	<i>AFx2</i>	<i>AFx3</i>	<i>BE</i>
Per-Flow	1.0	1.0	0.8	0.6	0.4
Per-Queue	0.2		0.4		1.0

The main goal of introducing packet marking as mentioned in Section 3.3 is to support service differentiation in the form of AF drop precedences and to improve overall buffer usage. Packet marking does not influence packet order, and packets belonging to the same traffic class will leave the router in the same sequence as they arrived.

The results shown in Figure 6 illustrate the differentiation among AF classes when FCFS or SCFQ scheduling is used. With SCFQ the scheduler completely dominates bandwidth allocation. Minimum-bandwidth guarantees for best-effort traffic can be given with both schedulers if packet marking is used. Without packet marking, best-effort traffic starts oscillating and loses reserved bandwidth even with SCFQ scheduling.

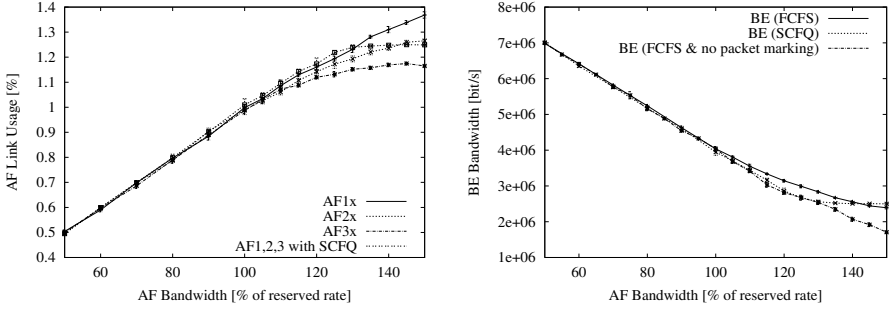


Fig. 6. Comparing AF and BE bandwidth for FCFS and SCFQ scheduling.

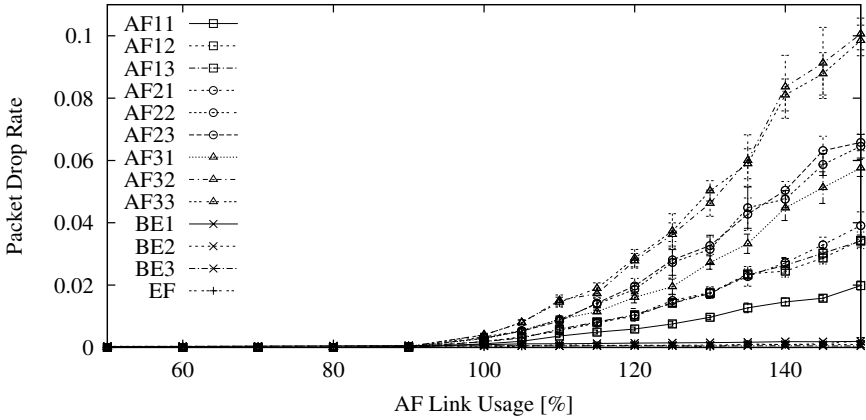


Fig. 7. Packet drop rate for various traffic classes using a FCFS scheduler.

With the given per-flow thresholds, every AF class is split into three drop precedences (Figure 7). Because of the high network load (when AF classes are sending more than 120% of the reserved rate), the buffer space of a router is almost completely filled up at any time, and the third per-flow threshold is too low to take effect. Nevertheless the Diffserv requirement of having at least two drop levels is satisfied. Relative service differentiation in terms of packet drop rates is clearly visible.

The packet delays are shown in Figure 8. For scheduling, SCFQ scheduler takes the packet arrival time as well as the number of packets being stored in a queue into account, whereas in a FCFS scheduler all queues experience the same average delay because FCFS cannot distinguish among the queues. Therefore with FCFS scheduling, the average delays for traffic classes other than best-effort are shifted towards the best-effort values when the actual AF bandwidth is lower than the reserved rate. However, delays have an upper bound given by the per-queue thresholds. This is not the case for FCFS scheduling, and only overall

buffer occupancy influences packet delay. To be more precise, decreasing per-queue thresholds would lower overall buffer usage because packets have already been dropped earlier to avoid congestion, and would have an equal effect on packet delays in all queues. On the other hand, we have seen that a WFQ scheduler imposes its fairness properties in a way that traffic differentiation is only feasible through static threshold settings. Although the average delay can be kept within an acceptable range, no significant delay differentiation can be realized with FCFS. Non-best-effort delays are always larger with FCFS for sources using less than their full share of bandwidth [18]. Here packet delay could be improved by using a virtual scheduling algorithm or a “weak” WFQ scheduler that allows higher-priority packets to bypass others.

5.2 AF: Gold, Silver and Bronze Services Using a Virtual Scheduler

To facilitate comparison of the results shown above using one scheduler with those described below, the same settings have been used but a virtual scheduler has been added. Again, the incoming traffic for the AF classes stems from CBR and multiple Pareto on/off sources. The AF sending rate range has been increased to 200% of the reserved rate to show that even with severe oversubscription tail drops are rare.

The above-described results show how packet drop rate differentiation can be achieved with FCFS while packet delay remains the same for all packets traversing the router. The virtual scheduler scheme has been introduced to overcome this weakness. Figure 9 shows the average packet delays for all queues. Surprisingly, what was better in terms of drop precedence in simple DRD now becomes worse in terms of delay: This means that the packet delay is shorter in higher-numbered AF classes and therefore AF3x has the best performance in terms of delay. As AF itself does not specify any particular relationship between AF per-hop-behaviors, the AF numbering introduced earlier will be kept. In addition it can be seen that the delay is bounded for each class separately. A intuitive explanation of this result is that DRD combined with a virtual scheduler will start dropping packets earlier in higher-numbered AF classes because

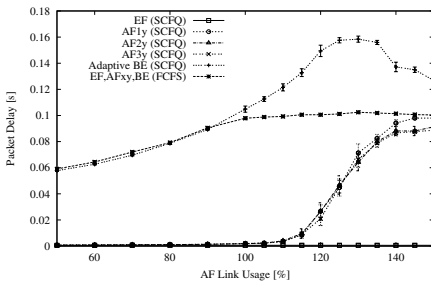


Fig. 8. Comparing packet delays without virtual scheduler.

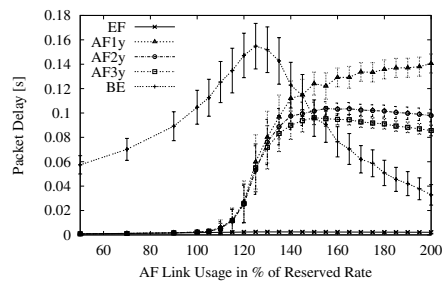


Fig. 9. Packet delays for different classes using a virtual scheduler.

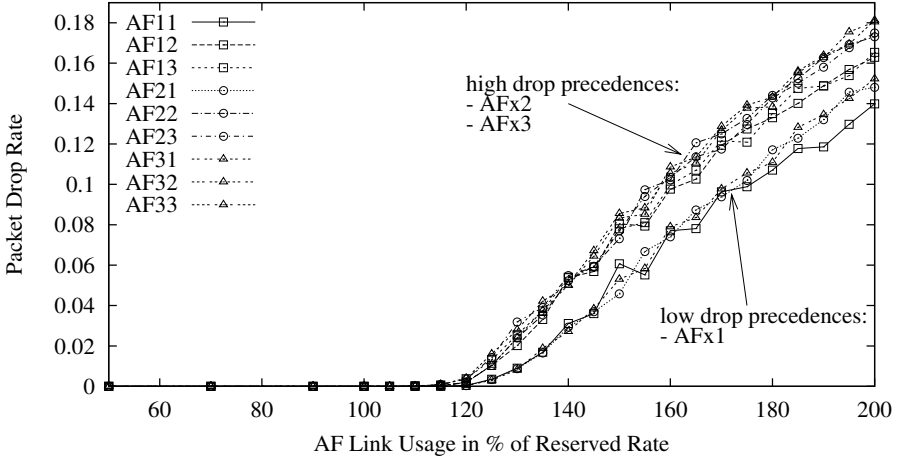


Fig. 10. Packet drop rate for all AF classes including the drop precedence.

of the higher DRD drop probability of those classes, whereas the real scheduler maintains the fair bandwidth allocation and therefore assures equal drop rates for equal incoming traffic.

In contrast to the simple DRD case, with FCFS packet drop rates are similar for all AF classes. Again only two levels of drop precedence in one class are visible (Figure 10). It will be shown later that this is only the case when all classes send at the same rate.

The results show that the virtual scheduler scheme is able to differentiate packet delays while giving a strict bandwidth guarantee according to the configured weight. Excess bandwidth is distributed according to the queue weights.

5.3 Delay Distribution

Packet delays are distributed approximately normally, as shown in Figures 11 and 12. The latter is a quantile-to-quantile plot, in which the straight line indicates a linear least-squares fit. The slightly S-shaped plots indicate that the distribution is peakier and has shorter tails than a normal distribution. This stems from the fact that delays cannot be negative and that overall buffer space is limited. The delay differentiation is due to the intrinsic behavior of the scheme rather than to sudden queue flushes or other undesired effects. In addition to the lower delay, the AF3y class also has a smaller delay variance, making it an attractive candidate for a “better than” service class.

5.4 Varying Incoming Traffic for One AF Class

In the preceding simulation, the AF sending rate has been varied for all AF classes. Now the rate is fixed to 100% of the reserved rate, and only one class at a time varies from 50% to 200%.

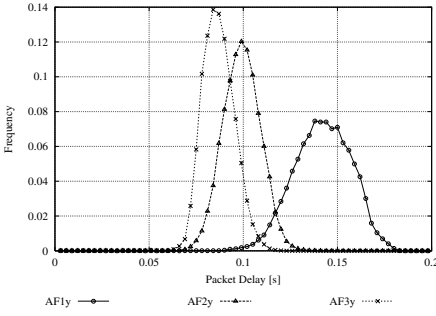


Fig. 11. Delay distribution for AF classes at 200% of reserved rate.

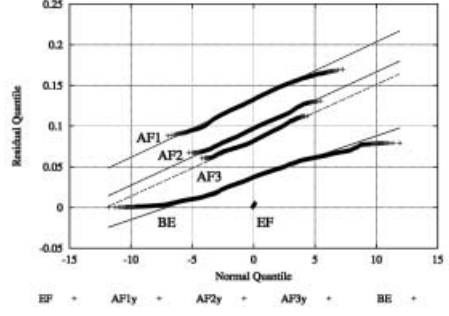


Fig. 12. Quantile-to-quantile plot for all delay distributions. The straight line indicates a linear least-squares fit.

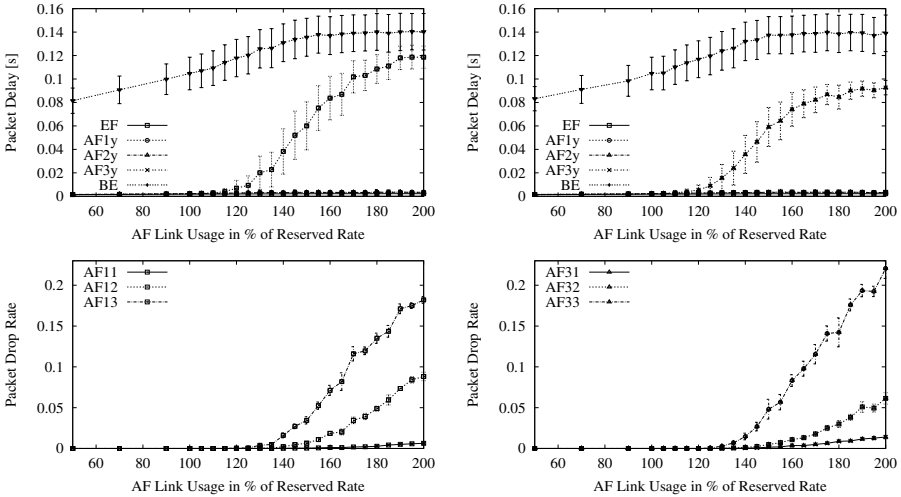


Fig. 13. Packet delay and drop rate while only one AF class varies.

Figure 13 show the packet delay and the drop rate when only one incoming rate (AF1y or AF3y) varies and confirms the results obtained in Section 5.1. In addition, a clear drop precedence differentiation between all three precedences, which in the previous simulation has disappeared, is now distinguishable again. This lets suggests that delay depends on the number of queues in the system. This has not been tested in this paper, and is left for future work.

5.5 Comparison to the Basic RED Algorithm

In a network environment with severe oversubscription and thus offered loads that by far exceed the transmission capacity, RED has turned out to be insufficient for efficient congestion indication if the number of TCP connections is high

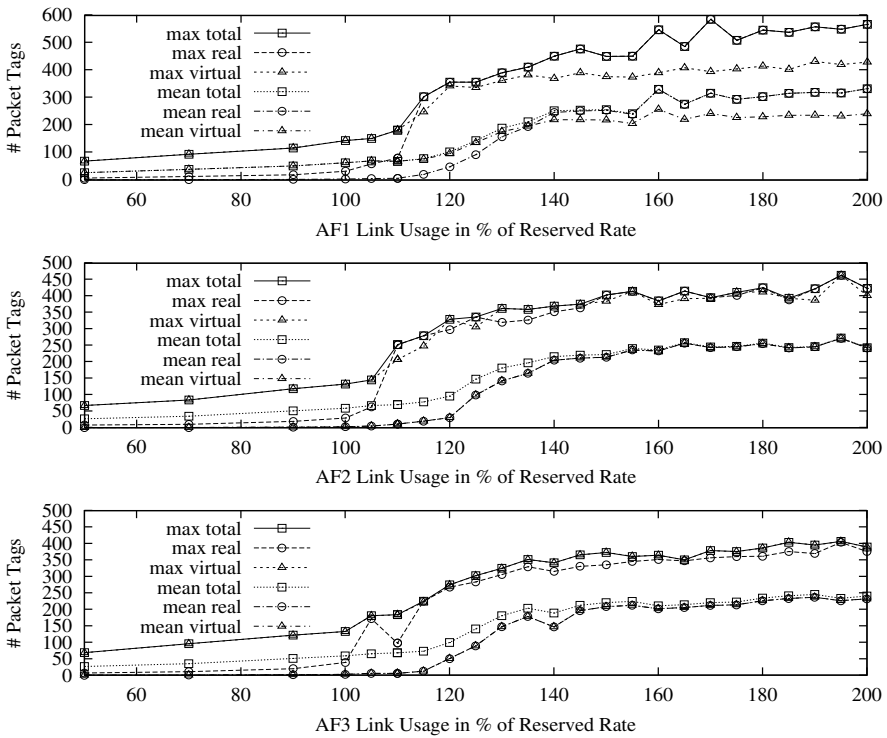


Fig. 14. Comparing packet tags for different AF under heavy load.

or traffic does not behave in a TCP-friendly way. Here we compare DRD congestion avoidance using a virtual scheduler with traditional RED. The total offered load for the simulation has been set to 120% of the available rate. Packet drops are counted during the 100 sec of simulation time. Table 3 shows that forced packet drops, known as tail drops, have been significantly reduced using DRD and virtual scheduler, and amount to less than 2% of all dropped packets. RED drops more than two thirds of all dropped packets because of buffer overflow. The conclusion is that DRD with virtual scheduling has a excellent potential for efficient, early congestion avoidance.

Table 3. RED vs. DRD with virtual scheduler.

	<i>Tot. offered Rate</i>	<i>Early drops</i>	<i>Tail drops</i>
RED	11.98 Mbit/s	44,639	92,459
DRD	11.71 Mbit/s	120,074	2,391

5.6 Packet Tags

As already mentioned, the new scheme needs more memory, mainly because additional packet tags have to be stored. First of all, it has to be shown that the number of additional tags is bounded. The set of tags in a queue i is given as T_i , and the subsets of real and virtual tags are T_i^r and T_i^v . The use of a second scheduler leads to an overall increase of packet tags in the system. The set of extra tags is written as $T_i^e = T_i^v \setminus (T_i^r \cap T_i^v)$. If only a real scheduler is used then $T_i = T_i^r$, otherwise, i.e. with a virtual scheduler, $T_i = T_i^r \cup T_i^v$. Figure 14 shows that $|T_{AF1} \setminus T_{AF1}^r| \rightarrow 0$ and $T_{AF1}^v \subset T_{AF1}^r$ under heavy load in the AF1 queue, whereas in the AF3 queue $|T_{AF3} \setminus T_{AF3}^v| \rightarrow 0$ and $T_{AF3}^r \subset T_{AF3}^v$. The consequence is that for the former the set of extra tags is $|T_{AF1}^e| \rightarrow 0$ and for the latter $|T_{AF3}^e| \rightarrow |T_{AF3}^v| \neq 0$, causing the increase in the total of packet tags. We found that $2|T^r| \geq |T|$ holds for all offered loads. As compared to a packet these tags are small, the impact on overall memory increase is justifiable.

6 Conclusion

In this paper we introduced a two-threshold-based buffer-management system that can be used for relative service differentiation in AF per-hop behaviors. The main new parts are the DRD congestion-avoidance scheme, internal packet marking, and a virtual scheduler. The DRD congestion avoidance scheme enables dynamic and relative service differentiation even with a simple scheduler such as FCFS. The fact that no delay differentiation is possible when used with FCFS led to the introduction of a virtual scheduler scheme. By means of simulations, it has been shown that a virtual scheduler is a robust management scheme for heavy and bursty traffic load. In conjunction with DRD, the scheme is able to perform relative delay differentiation of AF Diffserv per-hop behavior while guaranteeing minimum bandwidth and fair excess bandwidth allocation. The scheme avoids tail drops and, therefore, does not lead to TCP synchronization effects. Compared to other schemes, DRD with a virtual scheduler uses only few parameters (per-queue and per-flow thresholds, queue priority and queue weight) that are set at initialization time, and then requires no further tuning.

Packet marking is an important enhancement to flow-and-queue threshold-based buffer-management systems which allows the implementation of at least two drop precedences in a queue. In addition to optimizing overall buffer usage, packet marking is even necessary to avoid bursty packet drops. The influence of responsive and non-responsive flows in the same queue can have a significant impact on inter-flow fairness, but would exceed the scope of this paper and is left for future work.

Acknowledgments

The authors would like to thank Daniel Bauer and Robert Haas for their feedback and helpful discussions.

References

1. S.J. Golestani. A Self-Clocked Fair Queuing Scheme for Broadband Applications. *ACM Computer Commun. Rev.*, April 1994.
2. A.K. Parekh and R.G. Gallager. A Generalized Processor Sharing Approach to Flow Control. June 1993.
3. D. Stiliadis and A. Varma. Design and Analysis of Frame-Based Fair Queueing: A New Traffic Scheduling Algorithm for Packet-Switched Networks. May 1996.
4. J. C.R. Bennett and H. Zhang. WF²Q: Worst-case Fair Weighted Fair Queueing. 1996.
5. S. Floyd and V. Jacobson. Link-sharing and Resource Management Models for Packet Networks. *IEEE/ACM Trans. on Networking*, Vol. 3, No. 4, August 1995.
6. R. Pletka, P. Droz, and R. Haas. A New Buffer Management Scheme for IP Differentiated Services. Technical Report RZ 3216 (# 93262), IBM Research, Zurich Research Laboratory, March 2000.
7. S. Blake, D. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services, RFC2475, December 1998.
8. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group, RFC2597, June 1999.
9. S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Halder, M. Handley, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, and D. Zappala. Improving Simulation for Network Research. Technical Report 99-702b, University of Southern California, March 1999.
10. C. Drovolis, D. Stiliadis, and P. Ramanathan. Proportional Differentiated Services: Delay Differentiation and Packet Scheduling. *ACM Computer Commun. Rev. (SIGCOMM '99)*, September 1999.
11. F. Risso. Decoupling Bandwidth and Delay Properties in Class Based Queueing. *Dipartimento di Automatica e Informatica, Politecnico di Torino*, 2000.
12. S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *ACM Trans. on Networking*, August 1993.
13. R. Guérin, S. Kamat, and V. Peris. Scalable QoS Provision Through Buffer Management. October 1998.
14. D. Clark and W. Fang. Explicit Allocation of Best Effort Packet Delivery Service. *ACM Trans. on Networking*, Vol. 6, No. 4, August 1998.
15. V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB, RFC2598, June 1999.
16. J. Heinanen and R. Guerin. A Single Rate Three Color Marker, RFC2697, September 1999.
17. J. Heinanen and R. Guerin. A Two Rate Three Color Marker, RFC2698, September 1999.
18. A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of Fair Queueing Algorithm. *ACM Computer Commun. Rev.*, pp. 3-12, 1989.

Enhanced End-System Support for Multimedia Transmission over the Differentiated Services Network

Haisang Wu^{1*}, Huai-Rong Shao², and Xing Li¹

¹Dept. of Electronic Engineering
Tsinghua University, Beijing, P.R.China 100084

²Microsoft Research China
No. 49 Zhichun Road, Beijing, P.R.China 100080

Abstract. Proposed as a scalable solution to provide Quality of Service (QoS) networks, Differentiated Services (DiffServ) architecture enables service providers to offer each customer a range of services that are differentiated on the basis of performance. However, up to now little work discusses how to provide effective communication support for multimedia applications over the differentiated services network. In this paper we present a new enhanced communication approach at the end system to support distributed multimedia applications. Two new mechanisms are highlighted in our approach: enhanced communication stack to support differentiation within one flow, and the network-awareness provision for applications at the end system. Our approach improves the capability of both error resilience and flexible rate control for transmitting compressed multimedia bitstreams, particularly those using scalable coding technologies. We also develop an object-based video transmission system as an application instance to take advantage of the enhanced communication approach. Experimental results demonstrate the effectiveness of our proposed methods.

1. Introduction

Current Internet provides best-effort (BE) service to end-users and does not make any service quality commitment. However, most multimedia applications are sensitive to available bandwidth and delay experienced in the network. To satisfy these requirements, two frameworks have been proposed by IETF: the Integrated Services (IntServ) [1, 6], and the Differentiated Services (DiffServ) [10].

The IntServ model provides per-flow QoS guarantee and RSVP (Resource ReserVation Protocol) [5] is suggested for resource allocation and admission control. However, the processing load is too heavy for backbone routers to maintain states of thousands of flows. DiffServ model is designed to scale to large networks and gives a class-based solution to support relative QoS. The main idea of DiffServ is to minimize state and per-flow information in core routers by placing all packets in

* Corresponding author. E-Mail: hswu@public.bjnet.edu.cn Tel: (86-10) 62785814~525; Fax: (86-10) 62785933.

Mail address: Room 224 Central Building, Tsinghua University, Beijing 100084 P. R. China.

fairly broad classes at the edge of network. Core devices perform differentiated aggregate treatment of these classes based on the marking performed by the edge devices. A single byte in each packet is used to do this, called the DS byte (the Type of Service byte in IPv4 and the Traffic Class byte in IPv6), which can be set by the end station or ingress edge router to indicate the class of service desired. Since it is highly scalable and relatively simple, DiffServ model may be promising to dominate the backbone of the next generation Internet in the near future.

One of the most important research topics in multimedia networking is how various applications such as multimedia streaming and video conference can take full advantage of differentiation capability provided by DiffServ. Inherently, different kinds of information in the compressed bitstream may have different importance levels for the decoder to reconstruct the multimedia playback data. For instance, in MPEG4 shape and motion information is more important than texture for a P frame [15]. If shape and motion information is lost during transmission, the decoder cannot reconstruct the P frame successfully. However, if partial texture information is lost without the loss of shape and motion, it is still possible to reconstruct the P frame with acceptable quality [11, 15]. Another example is scalable coding such as layered approach [2, 4, 8] or Fine Granularity Scalability [14], in which enhancement layers are much less important than the base layer information. It seems that there is a natural mapping between different kinds of information with different classes of network packets, but unfortunately today's differentiation approaches can only support differentiation among flows, i.e., all packets belonging to the same flow have to be mapped to the same packet class. To satisfy the requirements of scalable multimedia, we propose an enhanced communication stack that can provide differentiation within one flow.

Currently, layered transmission and multicasting approach for scalable media can also differentiate base layers and enhancement layers by using multiple IP sessions. However, it is complicated for network and end system to maintain multiple sessions semantically bundled. Particularly, it is difficult to control the synchronization between these semantically bundled layers [2, 15]. On the other hand, with the functionality of differentiation within one flow, multiple-layer information can be transmitted within one IP session. Moreover, different kinds of information within one layer can also be easily differentiated in our approach.

Network-awareness support at the end system is another key issue for multimedia transmission over DiffServ network. With the development of network technologies and users' requirements, today's multimedia applications (such as MPEG4 applications) become more and more complicated. It is required that resources should be allocated and managed in a cooperative way, which means network resources can be dynamically coordinated among applications and flows within the applications adaptive to network fluctuations. We propose network-awareness support at the end-system to solve this problem successfully.

The rest of the paper is organized as follows. Section 2 presents the mechanism of differentiation within one flow. Section 3 discusses the network-aware support at the end system. In Section 4, we describe the experiments and the performance testing results of our approach. Section 5 introduces an application instance running on the proposed communication stack. Finally we conclude this paper.

2. Differentiation within One Flow

2.1. Design of the Enhanced Communication Stack

For complex multimedia applications such as MPEG4 programs, usually there are multiple objects of the same media or different media [15]. If these objects have different QoS requirements, generally each object is served by an individual IP session or even several sessions in the case of multiple layers. In general it is difficult for the end-system and network to maintain so many sessions for one application. By default, packets are marked based on a mapping from the service type associated with a flow, and all packets within one flow have the same marking value. However, if differential marking within a flow is supported, layers belong to the same object or different objects can be multiplexed into one IP session.

A possible problem resulting from differentiation within one flow is disorder of packets within the same flow, but this exists even without the function of differentiation within one flow because of the connectionless characteristic of IP protocol. To support the new differentiation functionality, an extension needs to be made on the protocol stack of the end-system.

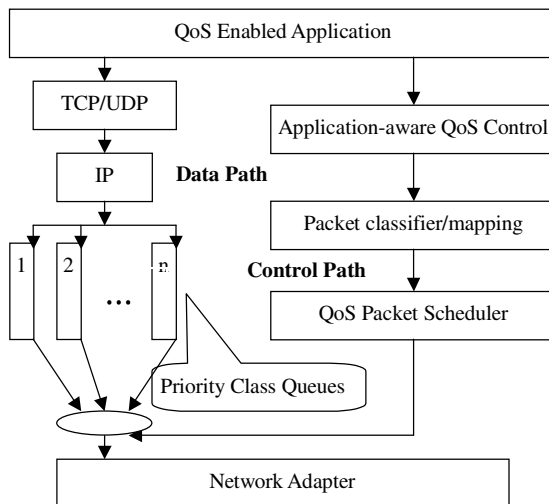


Fig. 1. Logical Architecture of the enhanced communication stack

We propose a new marker mapping mechanism in the host protocol stack to support differentiation within one flow. We introduce the multiple queue mechanism at the end-system and each queue buffers packets with a particular priority. When the IP header is added at the IP layer, the priority is mapped to the DSCP (DiffServ Code Point) byte.

To achieve our proposal, we intend to find the different priorities of packets according to DSCP value marked by QoS-aware applications, and then packets are classified into different classes according to specific rule. All this can only be performed through TCP/IP stack within operating system. Application should call the

kernel to set the DSCP value in IP packets header, and the kernel should finish all following queuing and scheduling.

After kernel sorts the packets generated by QoS-enabled applications into different classes according to their DSCP value, DiffServ scheduler in operating system sends out different classes of packets according to certain scheduling algorithm. Thus Priority Class Queues are used to implement packets scheduling.

The logical architecture of proposed system is shown in figure 1. It shows a host system, including applications, operating system and hardware (network adapter). In the operating system, the TCP/IP stack will perform the mechanism proposed and send the resulting packets into network adapter. It can be seen from Figure 1 that the data path and the control path are separated, and the control capability is improved.

2.2. Implementation of the Enhanced Communication Stack

We implemented our enhanced communication stack based on the open source code of Microsoft Research IPv6 stack release 1.4 (MSRIPv6 release 1.4).

Network protocols in Windows NT are dynamically loadable device drivers, much like any other device driver in Windows NT. It is possible to add a new protocol to the system by writing two new components: a kernel-level driver (tcpip6.sys) that exports the TDI interface and uses the NDIS interface, and a user-level helper (wship6.dll) to support access to the driver via sockets [7].

We made needed modification to MSRIPv6, most of which is done within these two components. Our main purpose is to add modules to set Traffic Class Value in IPv6 packet header and to queuing the packets according to their TC values. As a prototype, this would not add to the overhead of kernel.

3. Network-Aware End System

Figure 2 shows a framework example of the streaming server with intelligent resource control and management for multimedia applications. This framework considers the transmission of multiple-object video programs and other types of media such as audio and data. Each video object is compressed first and corresponding elementary stream is generated. Then information within each elementary stream is classified based on importance and assembled into packets with different DiffServ classes. Network Monitor is responsible for estimating the available network bandwidth dynamically through probing or feedback-based approach. Packet Forwarder forwards the packets to the network. We do not discuss these two blocks in details in this paper. The other functional components are described as follows.

- **Priority Mapping and Marking Agent**

This component is responsible for the interaction between applications and the DiffServ networks. It assigns DSCP marks to packets and maps them to the corresponding DiffServ classes.

- **Application Collaborator**

The Application Collaborator is responsible for resource coordination among multiple objects within one application and among multiple applications. It receives information from Application Profiles, Remote Users Interactions, and Network

Monitor to make the decision. In addition, the Application Collaborator tells how to map packet priorities from individual encoders into network classes. The receivers can interact with the server through user-level signaling.

- **Application Profiles**

This component records the semantic information of the applications such as which media and flows are included in an application and their relative importance levels.

- **Remote User Interactions:**

A user can interact with the video player or the server in several ways such as mouse clicking, mouse moving, fast forward, fast backward, object zoom-in, object zoom-out, add or delete. Some of these interactivity behaviors require dynamic adaptation of the bit rate of each video object and dynamic resource allocation coordination among multiple video objects. In object-based video multicast applications, different clients can have different views and interactions for the same video.

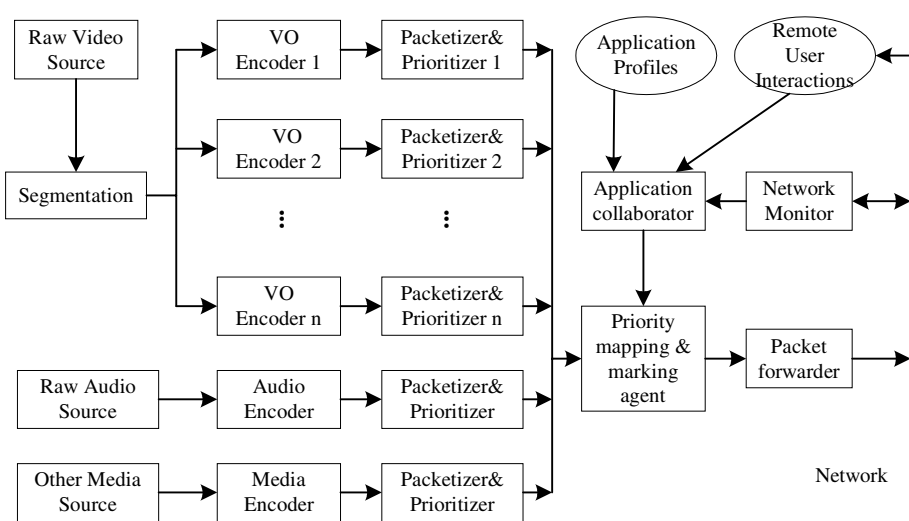


Fig. 2. Multimedia Communication Framework in the End-system

4. Performance Testing of the Enhanced Communication Stack

4.1. Testing Environment

The experimental testbed is shown in figure 3, which is an IPv6 testbed.

Box 1 is a PC installed with Win2000 operating system and MSIPv6 stack. The others are installed with Linux (Redhat) and UNIX (FreeBSD). All the boxes are dual-stack, i.e., IPv4 and IPv6 stacks. The “Internet” in the figure is a network comprised of several PCs, routers and switches. Box 4 is the default gateway of box

1, box 2 and box3. Box 5 is connected to box 4 via IPv6 over IPv4 tunnel, which is virtual link and different from physical links among the other boxes.

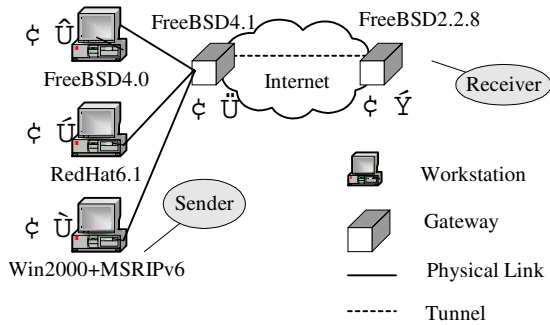


Fig. 3. Testing Environment

Box 1 is the sending machine and box 5 is the receiving machine in the testing. As we made our modification in sending module and leave the receiving module untouched, we only need to testify the sending functions, and the box 5 with UNIX installed should not affect the testing results.

With the original MSRIPv6 stack we are not able to set the value of TC field, but the TC value can be set with the modified stack. Thus the testing can be performed under four conditions. The composition is shown in table 1.

Table 1. Testing Condition

	Not setting TC (So TC=0)	Setting TC Randomly
Original Stack	•Condition •	Not available
Modified Stack	•Condition •	•Condition •

We define the condition 1 as “not setting TC value with the original stack”, and condition 2 as “not setting TC value with modified stack”, and so on. In the following paragraphs, we use “test 1” to represent “test carried out under condition 1”, and it is the same with “test 1” and “test 3”.

In the testing procedure, the sender, box 1 sends packets to the receiver, box 5 consecutively. Before sending, each packet’s TC field is marked with one number randomly chosen, and the packet content is filled with the same value as TC field in its header.

4.2. Experiment 1: Delay

One considered aspect is delay. We perform testing under the three conditions respectively, consecutively sending 40 packets to see the delay. The result is shown in figure 4 (left). At first the network link is with relatively light load. We find that the delay in test 1, 2 and 3 has no radical difference, and the delay time of each packet is around 1 millisecond except for a few bizarre points, which is presented in the curves.

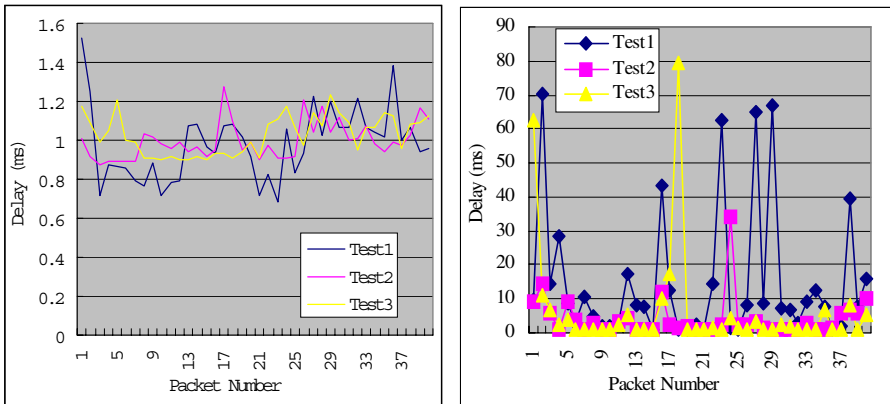


Fig. 4. Results of Delay Experimenting (Left: delay testing result without congestion Right: delay testing result in congestion condition)

Then we perform the same testing with heavy link load. Several other services assumed to consume much link resource such as FTP are started to create the link congestion. The testing result is shown in figure 4 (right). This time the delay of each packet and the variation within each condition are apparently much larger than before. But in the results we still cannot find the distinction among test1, test2 and test3.

4.3. Experiment 2: Packet Loss

The other aspect we consider is packet loss. The first problem that we care is whether adding queues in the stack to schedule different classes of packets would have negative effect on the performance of the stack. As described before, when applications do not set TC value, they leave this field to be zero, which is the default value in the original stack. Thus in case of not setting TC value, i.e., in condition 1 and condition 2 respectively, it would be more reasonable for us to compare results with different IPv6 stacks.

We create congestion with some traffic such as FTP to consume link resource. Some packet losses do occur, but very tiny, for our methods in testing environment is not enough to simulate the complex conditions of WAN. Both in test 1 and test 2 we tried 10 times to count packet loss in each time. The result is shown in figure 5 (left).

As the figure tells, there is no much difference in the results of test 1 and test 2, which may mean that queues added into the stack for class-scheduling do not impact too much negative effect on performance of the stack. But the testing environment and methods are too simple, and we cannot control sending speed of either the applications or the hardware (the network card). So the result in this figure does not mean that our modification would not affect the overall performance negatively in other conditions.

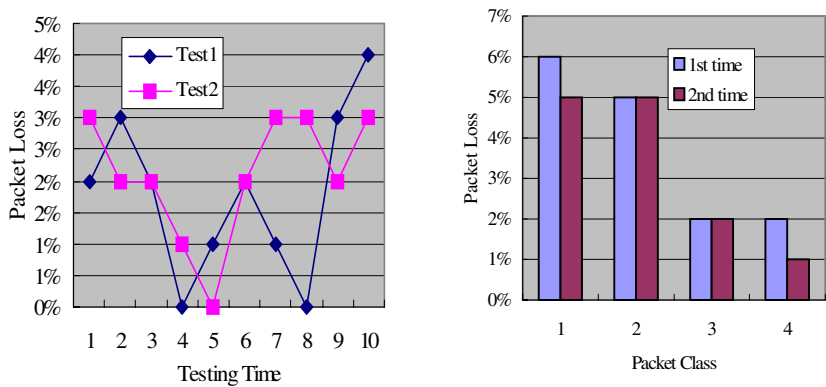


Fig. 5. Results of Packet Loss Experiment (Left: packet loss of different stacks Right: packet loss of different classes)

The other problem that we care is whether our mechanism can schedule different classes correctly, especially in congestion. We need to prove that with the scheduling mechanism, in congestion conditions packets with higher priorities can be sent out first. We performed two times of testing under condition 3. 100 Packets with TC value randomly set are sent out from box 1 to box 5 in a network link with very heavy load, and in box 5 the packet loss of each class is calculated. In this testing, we defined four classes; in which class 4 has the highest priority while class 1 has the lowest priority. Results of two times of testing are shown in figure 5 (right), from which we can see that packet losses of class 3 and class 4 are lower than those of class 1 and class 2. In the first time of testing, packet loss of class 1 is 6% and that of class 2 is 5%, but packet losses of class 3 and 4 are both 2%. In the second time, packet losses of class 1 and 2 are both 5%, but that of class 3 is 2% and 1% of class 1. This proves that the scheduling mechanism is effective in packet scheduling.

5. Application Instance Running on the Enhanced Communication System

It can be seen from Section 4 that the enhanced communication stack can differentiate different classes of packets without much performance impairment. However, the enhanced stack can greatly benefit the transmission of scalable multimedia.

We implemented a MPEG4 video streaming system and run it on the proposed enhanced communication stack within a simulated DiffServ network. In this system, we proposed a new bitstream classification, prioritization, and packetization scheme in which different types of data such as shape, motion, and texture are re-assembled, assigned to different priority classes, and packetized into different classes of network packets provided by DiffServ. Taking advantage of the enhanced communication stack, our scheme distinguishes not only different kinds of frames, but also different types of information within the same frame. Readers can refer [15] to obtain details of our new transmission scheme. Besides our scalable transmission approach, for the

sake of comparison we also implemented the traditional approach in which the bitstream is packetized with no information re-organization/prioritization and all packets have a fixed size (600bytes). Figure 6 and 7 show the comparison results example of our approach and the traditional approach for Bream. It can be seen that our proposed transmission approach based on the enhanced communication is much better than the traditional one.

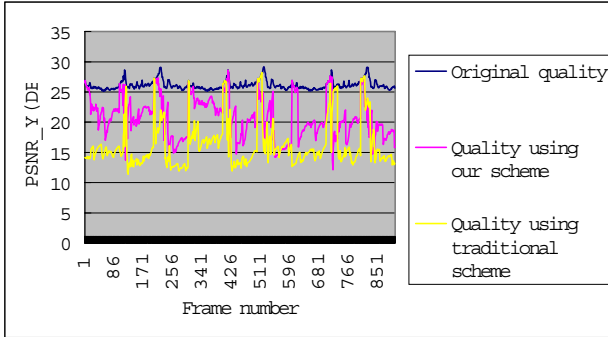


Fig. 6. Video quality comparison for Bream at 11.7% packet loss rate (actual bit rate=168kbps; original bit rate=187kbps).

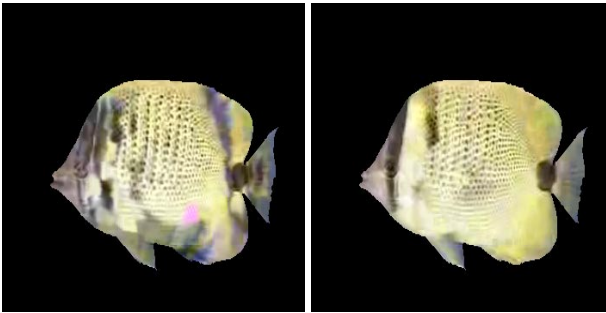


Fig. 7. Video frame example (Number: 150) Left: traditional approach Right: the proposed scheme.

6. Conclusion

In this paper we presented a new enhanced communication approach at the end system to support distributed multimedia applications. Two new mechanisms were highlighted in our approach: enhanced communication stack to support differentiation within one flow, and the network-awareness provision for applications at the end system. Our approach improves the capability of both error resilience and flexible rate control for transmitting compressed multimedia bitstreams, particularly those using scalable coding technologies. We have implemented a prototype and used an experimental testbed to study the effect of our approach. Experimental results

demonstrate that our approach can be an effective means for packet classification and scheduling for multimedia transmission.

References

- [1] S. Shenker, R. Braden, D. Clark, "Integrated services in the Internet architecture: an overview", RFC 1633, June 1994.
- [2] McCanne, "Receiver-driven layered multicast", SIGCOMM Symposium on Communications Architectures and Protocols, Palo Alto, California, August 1996.
- [3] D. Hoffman and M. Speer, "Hierarchical video distribution over internet-style networks", ICIP'96, Lausanne, Switzerland, September 1996.
- [4] X. Li, M.H. Ammar, and S. Paul, "Layered video multicast with retransmission (LVMR): Evaluation of error recovery", Proceedings of NOSSDAV, May 1997.
- [5] R. Braden, L. Zhang, and S. Berson, et al, "Resource Reservation Protocol - Version 1 Functional Specification", RFC 2205, September 1997.
- [6] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997.
- [7] Richard P. Draves, Allison Mankin, Brian D. Zill, "Implementing IPv6 for Windows NT", Proceedings of the 2nd USENIX Windows NT Symposium, Seattle, WA, Aug. 3-4, 1998.
- [8] D. Sisalem, "QoS Control using Adaptive Layered data transmission", IEEE International Conference on Multimedia Computing and Systems, 1998.
- [9] K. Nichols, S.Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC2474, December 1998.
- [10] S. Blake, M. Carlson, and E. Davies, et al, "An Architecture for Differentiated Services", RFC 2475, December 1998.
- [11] "MPEG-4 video verification model version 13.0", ISO/IEC JTC1/SC29/WG11/N2687, March 1999.
- [12] B. Carpenter, D. Kandlur, "Diversifying Internet Delivery", IEEE Spectrum, Nov. 1999.
- [13] M. Hemy, U. Hengartner, P. Steenkiste, T. Gross, "MPEG System Streams in Best-Effort Networks", Packet Video'99, New York.
- [14] S. Li, F. Wu, Y. Zhang, "Study of a new approach to improve FGS video coding efficiency", ISO/IEC JTC1/SC29/WG11/m5583, December 1999, Maui.
- [15] H. Shao, W. Zhu and Y. Zhang, "User-aware Object-based video multicast over the Internet", 10th International Workshop for Packet Video, May, 2000, Sardinia, Italy.

Investigations into the Per-Hop Behaviors of Diffserv Networks

Zesong Di and H.T. Mouftah

Department of Electrical and Computer Engineering
Queen's University, Kingston, Ontario, Ccanada K7L 3N6

Abstract. This paper presents investigations into the three kinds of per-hop behaviors of Diffserv networks. The investigations were carried out using Queen's University IP Simulator v2.0 (QUIPS-II). In terms of packet delay and packet drop rate, the per-hop behaviors were observed under various traffic and network parameters. The results show that the per-hop behaviors can efficiently differentiate between service levels on the Internet.

1 Introduction

Differentiated Services (Diffserv or DS) [2,4] architecture is proposed to provide scalable service discrimination on the Internet. It has recently become a promising method to address IP Quality of Service (QoS) issues. Instead of per-flow treatment in RSVP (Resource reSerVation Protocol) [1], Diffserv provides QoS to each packet in the traffic stream. It uses the DS field (which was the TOS octet) in the IP header to distinguish QoS requirements of each packet. At the network boundary, IP packets are classified to a smaller number of aggregated flows, based on setting bits in the DS field. Within the core of the network, each aggregated flow is forwarded according to a particular forwarding treatment, called per-hop behavior or PHB, which is associated with the DS codepoint. In this way, different traffic can obtain different QoS treatments. By pushing complexity to the network edge, this approach requires neither signaling nor per-flow state maintenance within the core of the network. Thus it greatly reduces the network overhead, which in turn increases its potential for scalability.

Sharing many features of the IETF proposals, we divide the Diffserv PHBs into three kinds: the Expedited Forwarding (EF) PHB [3], the Assured Forwarding (AF) PHB [5], and the Best-Effort (BE) PHB. The three forwarding treatments can build three corresponding Internet services: the Premium service, the Assured service, and the default best-effort service. The Premium service is a low loss, low latency, low jitter, assured bandwidth service. Such a service can be used to create a "virtual leased line", which greatly reduces the cost of building a separate network. The Premium traffic is characterized by a desired peak rate for a specific flow. The user contract with the network is not to exceed the peak rate. The network contract is that the contracted bandwidth will be available when traffic is sent. The Assured service provides a customer with the

assurance of a minimum throughput, even during periods of congestion. It allows him to consume more bandwidth when the network load is low. The minimum throughput equals to the subscribed minimum rate, which is called the target rate. This kind of service provides a “better effort” service by controlling the drop preference of packets at the time of congestion. As we already know, the default best-effort service has no QoS meaning. Its “first come first serve” rule provides no guarantee to any of loss, latency, jitter or throughput of a traffic flow.

The objective of this paper is to investigate the three PHBs in a Diffserv environment, especially the EF PHB and the AF PHB. In order to carry out the investigation, we have developed the Queen’s University IP Simulator v2.0 (QUIPS-II). By simulation, we can evaluate the performance of the PHBs. The rest of the paper is organized as follows. Section 2 presents the implementation of Diffserv in QUIPS-II. Section 3 describes the simulation experiments and results of the PHBs in a Diffserv network model. Finally, Section 4 provides the summary of the paper.

2 Implementation of Diffserv

The implementation of Diffserv in QUIPS-II is present in each network node. This is because in the Diffserv Internet, it is nodes, or routers, who are responsible for handling packets on different traffic flows and applying different treatments to them. The nodes can be separated into two categories: the edge nodes and the interior nodes. Both types of node are able to forward packets based on the DS codepoints which are associated with the PHBs. Moreover, the edge nodes are also responsible for traffic conditioning when traffic is entering or leaving a DS domain. In QUIPS-II, apart from the default BE PHB, each node deploys both the EF PHB and the AF PHB. Here, we do not elaborate the AF PHB to multilevel behaviors as defined by IETF, for the simplification and approximation of the simulation model.

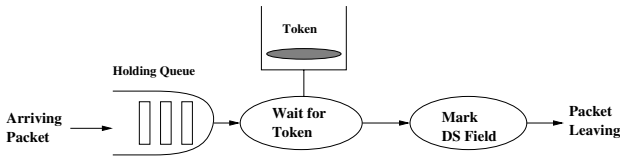


Fig. 1. The structure of a Premium marker

When a packet arrives at the input interface of an ingress edge node, it is first classified and then sent into a marker. Each traffic flow has an individual marker to treat its packets. Hence there are three kinds of markers in the edge node, i.e. Premium marker, Assured marker and best-effort marker. A Premium marker, as shown in Fig. 1, is actually a token bucket which is configured with the peak

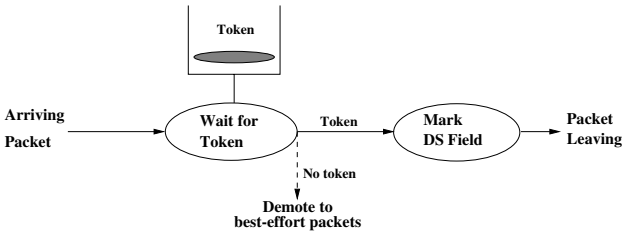


Fig. 2. The structure of an Assured marker

rate of the Premium flow. An Assured marker, as shown in Fig. 2, is also a token bucket which is configured with the target rate of the Assured flow. A best-effort marker, however, is only a marking mechanism without a bucket regulator. As soon as a Premium packet sees a token present, it is forwarded after having its DS field marked. If no token is available, the Premium packet will be held until a token arrives. Once a Premium flow bursts enough to overflow the holding queue, the surplus packets will be dropped. When an Assured packet emerges from its Assured bucket, it is marked as IN-profile. The non-conforming Assured packets, however, are not discarded immediately. They will be demoted to best-effort packets. In addition, all the best-effort packets are marked as OUT-of-profile and sent into the network. After the traffic conditioning, a marked packet is added to a certain behavior aggregate and sent to the output interface of the edge node for further forwarding.

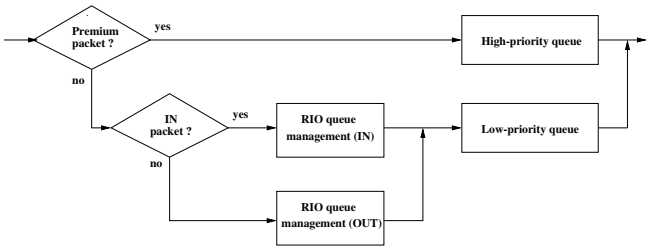


Fig. 3. The output interface of each node

Each output interface of a node employs a two-level priority queue mechanism. The Premium packets are assigned to the high-priority queue, while the Assured and the best-effort packets are assigned to the low-priority queue. The high-priority queue has a simple “non-preemptive” priority over the low-priority queue, which ensures that the Premium packets are sent first. While the Premium traffic keeps not oversubscribing, it will see no or very small queue. The priority queue mechanism can provide the EF PHB to a Premium flow. A RIO active queue management mechanism is implemented on the low-priority queue. RIO, which will be introduced next, is RED [6] modified to handle a mix of IN

and OUT packets. The active queue management mechanism can support the requirement of the AF PHB. Fig. 3 is a block diagram of the output interface for all the nodes in the network.

The RIO algorithm can be viewed as a combination of two RED algorithms. When a packet arrives, RIO determines if the packet is IN or OUT. If it is an IN packet, the router only calculates the average queue size for the IN packets (avg_{q-IN}). Otherwise, the algorithm calculates the average queue size taking into account all packets ($avg_{q-TOTAL}$), regardless of their markings. The probability of discarding an IN packet depends on avg_{q-IN} , whereas the probability of discarding OUT packets depends on $avg_{q-TOTAL}$. Essentially, RIO discards OUT packets first whenever it detects an emerging congestion. If the congestion persists, RIO will discard all the OUT packets and then begin to discard IN packets, in the hope of controlling the congestion.

3 Simulation Experiments and Results

We simulated a Diffserv network as shown in Fig. 4. The network consists of two edge nodes each with two-level priority queues. One edge node performs traffic conditioning to the incoming traffic, while the other does not do so to the outgoing traffic. In this study, we ignored the effects of interior nodes on PHBs in order to simplify the model.

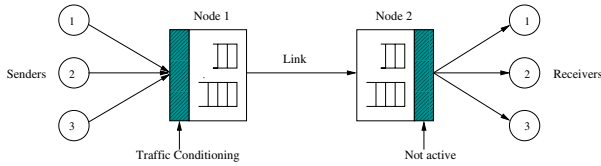


Fig. 4. The network model

There are three types of traffic coming into the network: the Premium, the Assured and the best-effort. They compete for a bottleneck link with bandwidth of 45Mbps (5.625Mbps). As in the example of [7], the assignment of the link bandwidth to each Diffserv class is 20% for the Premium, 40% for the Assured, and the remaining 40% for the best-effort. In the simulation, traffic is generated by Constant Bit Rate (CBR) sources, with a variation of $\pm 10\%$ of its source speed. For the packet length, it is set to 1500 bytes for all simulation runs. As the baseline situation, the size of the Premium holding queue in the edge node and that of the high-priority queue are both set to 5 packets, while the size of the RIO queue is set to 500 packets. As we already know, there are four parameters in RED, namely the minimum threshold min_{th} , the maximum threshold max_{th} , the maximum drop probability max_p and the queue weight w_q . Hence, RIO has two sets of parameters, i.e. $\langle min_{th-IN}, max_{th-IN}, max_{p-IN}, w_{q-IN} \rangle$ for the IN packets, and $\langle min_{th-OUT}, max_{th-OUT}, max_{p-OUT}, w_{q-OUT} \rangle$ for the OUT packets.

the OUT packets. In the network model, the RIO parameters are chosen as $\langle 225, 400, 0.02, 0.004 \rangle$ for the IN packets and $\langle 200, 400, 0.05, 0.004 \rangle$ for the OUT packets.

In the experiments, the Diffserv PHBs were observed in terms of packet delay and packet drop rate, since the two aspects make the PHBs fundamentally different. The parameters to be varied in the simulation model are packet length, traffic load, and service allocation. Note that in each run of simulation, only one parameter is changed while the others remain the same as defined in the baseline case. Some other performance evaluation of Diffserv can be found in [7,8].

3.1 Effect of Varying the Packet Length

The loads of EF, AF and BE traffic classes in this experiment are all set to 1. Here, the traffic load is defined by the proportion of the incoming traffic rate to the amount of bandwidth assigned to this service. For instance, in the experiment, the AF traffic load of 1 represents the ratio of the incoming AF traffic rate of 2.25Mbps to the assigned AF bandwidth which is 40% of the 5.625Mbps total link bandwidth.

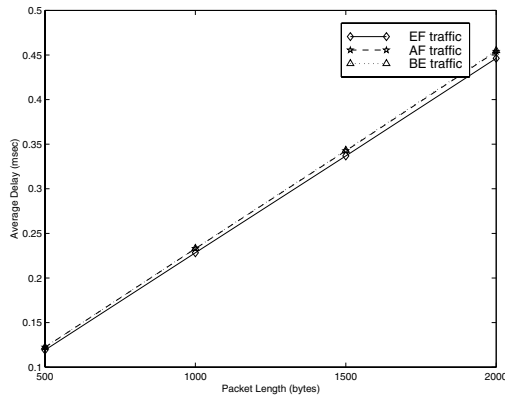


Fig. 5. Delay behaviors as a function of packet length

Fig. 5 illustrates a comparison of the packet delays experienced with different packet lengths. At any packet length, the delays of the three traffic classes are almost the same because no packet waits more than a packet-time inside its queue. When the packet length increases, the three delay curves will increase linearly and synchronously. The reason is quite straightforward. The longer a packet, the longer transmission time it takes to traverse the network.

For the three traffic classes, none of them experience packet loss. This is because under this network condition, there are no traffic congestion in the network. Moreover, varying the packet length does not give any effect to the packet drop rate.

3.2 Effect of Varying the AF Traffic Load

In this experiment, we observe the EF, AF and BE behaviors in response to the change in AF traffic load. Here, only the amount of AF traffic oversubscribes its reserved bandwidth, the EF traffic load and BE traffic load are kept at 1.

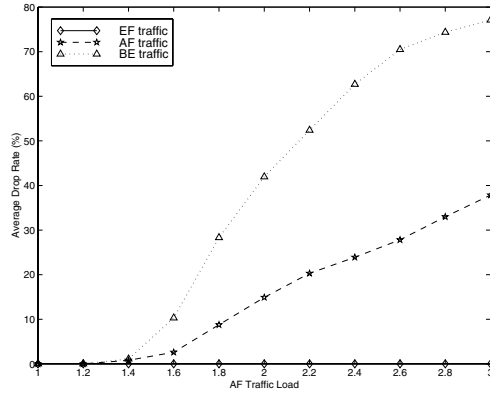


Fig. 6. Packet drop behaviors as a function of AF traffic load

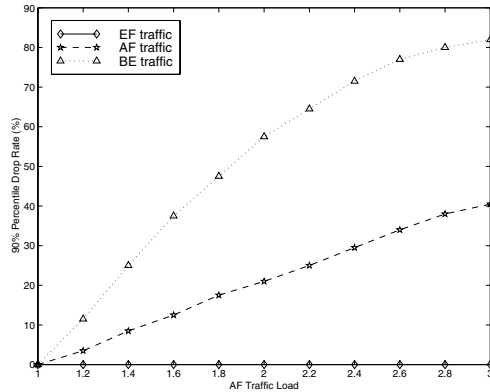


Fig. 7. Packet drop behaviors as a function of AF traffic load (steady state)

Fig. 6 and Fig. 7 show packet drop behaviors as a function of AF traffic load. As shown, the EF drop rate is 0 because EF traffic has got just enough bandwidth. However, the AF and the BE drop rates increase while the AF traffic load increases. The BE drop rate is larger than the AF drop rate since BE packets are dropped preferentially with respect to AF packets. In Fig. 6, when the AF traffic load is smaller than 1.2, the average drop rates of both AF

and BE are very small (negligible). This is because we have chosen a relatively long queue and large thresholds to handle AF and BE packets, which prevent packet dropping when the queue expands slowly. If observed at the steady state, as shown in Fig. 7, the two drop rate curves begin to rise when the AF traffic load is just over 1.0.

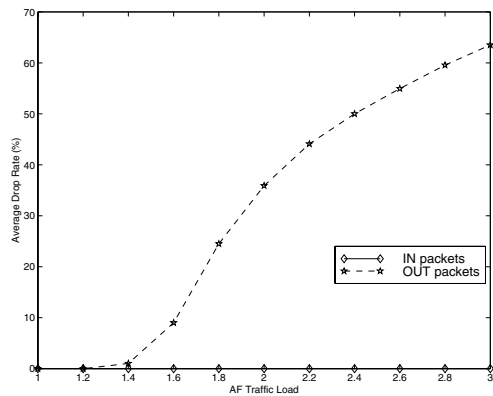


Fig. 8. Packet drop behaviors of IN and OUT packets

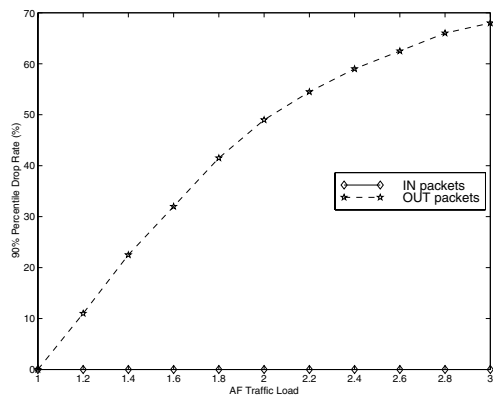


Fig. 9. Packet drop behaviors of IN and OUT packets (steady state)

The different packet drop behaviors of IN and OUT packets are shown in Fig. 8 and Fig. 9. As introduced before, IN packets are the conforming AF packets, whereas OUT packets are the non-conforming AF packets plus all the BE packets. IN packets should be transmitted through the network with no or little dropping, whereas OUT packets have no such assurance. The two figures reflect exactly such behaviors.

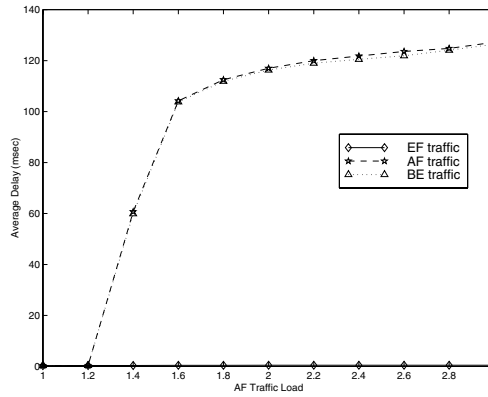


Fig. 10. Delay behaviors as a function of AF traffic load

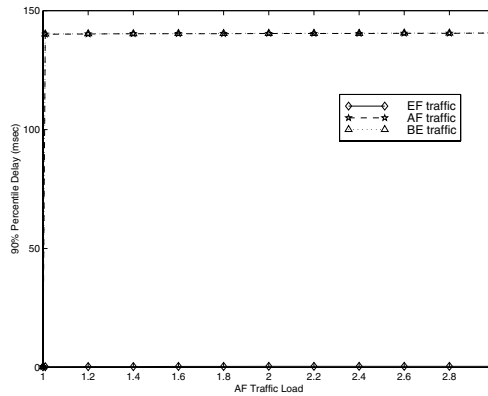


Fig. 11. Delay behaviors as a function of AF traffic load (steady state)

Fig. 10 and Fig. 11 compare the different delay behaviors of the three traffic classes. In Fig. 10, when the AF traffic load increases from 1.0 to 1.2, the average delays of AF and BE do not increase significantly. However, when the AF traffic load is higher than 1.2, the two delay curves begin to rise radically. In this situation, the low-priority queue becomes full quickly, which brings huge increase to both the AF average delay and the BE average delay. At an AF traffic load larger than 1.6, the two curves become smooth again with slow increasing trends. Drawn at the 90% percentile, Fig. 11 shows that when the AF traffic load is just over 1.0, both the AF delay and the BE delay jump from 0.344 millisecond up to 141 millisecond. No matter how we increase the AF traffic load afterwards, the two delay results do not change. This is because at the steady state, the low-priority queue has been already full. Therefore, all the new incoming AF and BE packets experience the same delay.

The EF delay curve in Fig. 10 is actually an increasing curve, although it is not shown very clearly in the graph. As we know, if two packets from two different priority queues emerge at the same time, one packet has to wait while the other gets the chance to occupy the common link. So, when the density of AF traffic increases, the AF packets will have more chance to occupy the common link, bringing the EF packets some extra delays. However, the increase is very small, with a maximum bound of one packet-time.

3.3 Effect of Varying the EF Traffic Load

Here, we observe the performance of EF, AF and BE traffic classes under different EF traffic loads. Fig. 12 and Fig. 13 show the performance results.

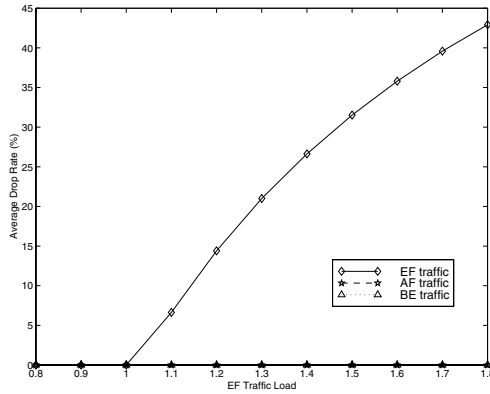


Fig. 12. Packet drop behaviors as a function of EF traffic load

In Fig. 12, neither AF traffic nor BE traffic drops packets. However, the EF drop rate increases almost linearly as its load increases. This is fairly straight forward. First, both the AF traffic load and the BE traffic load are kept at 1. Second, when the EF traffic load is larger than 1, the EF holding queue in the edge node will become full and begin to drop packets. The larger the EF traffic load, the more EF packets it drops.

The delay results are shown in Fig. 13. For the AF and BE delays, they remain at about 0.344 millisecond and are not affected by the EF traffic load. However, when the EF traffic load is just over 1.0, the EF delay jumps vertically from 0.337 millisecond to about 6.40 millisecond then saturates. The reason for the jump is that each EF packet has to wait a fixed amount of time inside its holding queue at the network edge.

3.4 Effect of Varying the EF Traffic Allocation

In this experiment, the effect of varying the EF traffic allocation on the delay behavior is investigated. We vary the EF reserved bandwidth from 10% of the

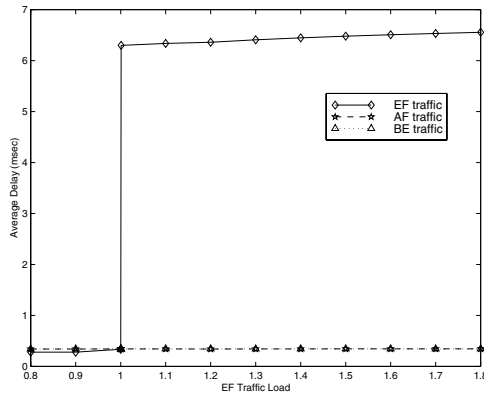


Fig. 13. Delay behaviors as a function of EF traffic load

link bandwidth to 50%, while keeping the AF reserved bandwidth at 40%. The remaining bandwidth is left for BE traffic. The loads of AF, EF and BE traffic flows are all set to 1. Fig. 14 shows the delay behaviors of the three traffic classes. While the EF traffic allocation increases, the EF delay decreases until it reaches a saturation of about 0.278 millisecond. However, the AF delay and the BE delay do not change significantly. The reason is that, when the EF traffic allocation increases, the EF packets will have more chances to be transferred by competing with other packets.

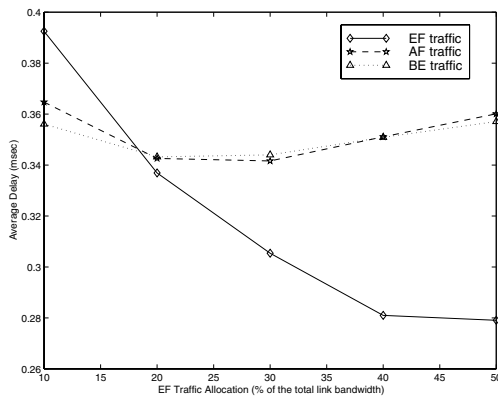


Fig. 14. Delay behaviors as a function of EF traffic allocation

4 Summary

In this study, we used QUIPS-II simulator to investigate the Diffserv PHBs in an IP network environment. By varying parameters in the simulation model, we observed and compared the performance of three different kinds of PHBs. The results show that the Diffserv PHBs can efficiently differentiate between service levels on the Internet.

References

1. Zhang L., Deering S., Estrin D., Zappala D.: RSVP: A New Resource ReSerVation Protocol. IEEE Network Magazine, September 1993, pp. 8-18
2. Black D., Blake S., Carlson M., Davies E., Wang Z., Weiss W.: An Architecture for Differentiated Services. Internet RFC2475, December 1999
3. Jacobson V., Nichols K., Poduri K.: An Expedited Forwarding PHB. Internet RFC2598, June 1999
4. Nichols K., Jacobson V., Zhang L.: A Two-bit Differentiated Services Architecture for the Internet. Internet RFC2638, July 1999
5. Heinanen J., Baker F., Weiss W., Wroclawski J.: Assured Forwarding PHB Group. Internet RFC2597, June 1999
6. Floyd S., Jacobson V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, August 1993, pp. 397-413
7. Naser H., Garcia A., Magd O.: Voice over Differentiated Services. Internet Draft, draft-naser-voice-diffserv-eval-00.txt, December 1998
8. Hou M., Mouftah H.: Performance Evaluation of Premium Service. Internet Draft, draft-hou-diff-premium-eval-00.txt, June 1999
9. Di Z., Mouftah H.: Performance Evaluation of Per-Hop Forwarding Behaviors in the Diffserv Internet. Proceedings IEEE International Symposium on Computers and Communications (ISCC'2000), Joan Les Pins, France, July 2000, pp. 334-339

An Adaptive Bandwidth Scheduling for Throughput and Delay Differentiation

Huu-Thanh Nguyen and Helmut Rzehak

University of Federal Armed Forces Munich,
Werner-Heisenberg-Weg 39, 87755 Neubiberg, Germany
{thanh,rz}@informatik.unibw-muenchen.de

Abstract. In the article, we proposed a new scheduling architecture based on rate-proportional servers, which is able to support different services in the DiffServ model by using a single service discipline. The scheduling algorithms allow to reduce the implementation complexity at the network node in the sense that there is no need to deploy a multi-level scheduling architecture for multiple types of services and for link sharing. The network operator is able to provide not only delay differentiation but also other services, such as bandwidth reservation, link-sharing or traffic engineering, etc.

1 Introduction

A few years ago the essential Internet applications were mainly such elementary services as e-mail, Web-surfing or file transfer. In contrast, users today expect that Internet service providers (ISP) offer different services as well as price patterns so that they can choose the one appropriate for them. Consequently, service providers have to not only provision higher capacity links, but also need to introduce more sophisticated network architectures, which can satisfy varied requirements of different customers.

An evolutionary approach to provide service differentiation in the Internet is the DiffServ model [2]. The main goal of DiffServ is to overcome scalability problems in the IntServ model [1]. Instead of providing service to individual flows, DiffServ supports only a limited number of classes of service. Flows belonging to the same class receive the same service from the network. A router in the DiffServ model has only to focus on traffic aggregates of classes of service, thus reducing complexity. In this work, we focus on a research direction of the DiffServ model called *Relative Service Differentiation* (RSD). In the Relative Service Differentiation approach, traffic from a higher priority class receives better (or at least not worse) services than lower classes in terms of queueing delays and packet losses [3,4,5,6]. Services offered in RSD networks are per-hop rather than on an end-to-end basis. A well-known model for the Relative Service Differentiation is the *Proportional Differentiation Model*. Quantitatively, service offered by class i in the Proportional Differentiation model is defined as follows [3]. Let q_i be the performance measure for class i , e.g., delay or loss, then the following equation is applied for all classes of service:

$$\frac{q_i}{c_j} = \frac{c_i}{c_j} \quad \forall i, j \in \{1, \dots, N\} . \quad (1)$$

where $c_1 < c_2 < \dots < c_N$ are the generic *quality differentiation parameters* and are selected by the network operator. Early work by Dovrolis, et al. [3] outlined definition and main issues of relative service differentiation, it also introduced two scheduling algorithms for *delay differentiation*, namely the *Backlog Proportional Rate* scheduler (BPR) and the *Waiting Time Priority* scheduler (WTP). In our paper, we argue that DiffServ networks must facilitate not only delay differentiation but also other service performance at the same time. Besides delay, customers, who use the DiffServ network for data transmission for example, would require such another service as throughput differentiation. Furthermore, users from different organizations or groups of service, that is, groups demanding delay performance vs. groups demanding throughput performance, require that the DiffServ network provide a link sharing mechanism to control the load distribution between them. In the article we present another WFQ-like service scheduling that can maintain service differentiation as accurate as that under WTP and provides a mechanism for throughput differentiation and link sharing at the same time. The rest of the article is organized as follows. Section 2 discusses some remaining issues on the existing schedulers and the main reasons that motivate us to develop a delay and throughput differentiation model, which is integrated in a single scheduler. Section 3 describes our scheduling algorithms. Performance study is shown in Section 4. The last Section concludes the work and outlines further possible research on the direction.

2 Previous Work on Relative Differentiation Service Schedulers

Until now, there are some approaches for the Proportional Differentiation Model [3,4,5,6]. In the article, we focus on BPR and WTP schedulers proposed in [3]. In our views, besides delay that is important for real-time applications, throughput is another fundamental performance metric for best-effort services. Moreover, link sharing is an essential mechanism that allows the network operator to effectively control traffic load between service types, protocol families, multiple agencies or carry out traffic engineering. Thus, a DiffServ network should support different services simultaneously, such as delay and throughput differentiation as well as perform link sharing. Starting from this point, we are now going to examine the advantages and disadvantages of WTP and BPR schedulers and how they can support various service performance.

WTP is a priority scheduler in which the priority of a packet increases proportionally with its waiting time. The priority of a packet in queue i at time t is defined as follow:

$$p_i(t) = w_i(t)s_i . \quad (2)$$

where $w_i(t)$ is the waiting time of packet i at time t , $\{s_i\}$ is the set of *Scheduler Differentiation Parameters*. The WTP scheduler, on the one hand, is found to be consistent in approximating the proportional delay differentiation model defined in Equation 1 under different load condition and traffic patterns [3]. On the other hand, by decoupling delay from service rate, it is difficult to maintain other types of service performance, e.g., throughput differentiation etc., at the same time without using a multi-level scheduling architecture. For example, in WTP a user cannot guarantee that he gains more bandwidth by using higher class of service. Thus, WTP scheduler is consistent in delay differentiation but not in bandwidth differentiation.

In contrast with the Waiting Time Priority scheduler, BPR relies on a property of Generalized Processor Sharing (GPS) systems [7] that delay of a packet depends on the rate allotted to the packet's session and the queue backlog of that session at the time the packet arrives. BPR reallocates rates to its classes of service proportionally to their backlog. Let $r_i(t)$ be the service rate assigned to queue i at time t , $q_i(t)$ be the queue i backlog at time t . For two back-logged queues i and j , the service rate allocation in BPR satisfies the proportionality constraint:

$$\frac{r_i(t)}{r_j(t)} = \frac{s_i q_i(t)}{s_j q_j(t)} . \quad (3)$$

$$\sum_{i=1}^N r_i(t) = R . \quad (4)$$

where $\{s_i\}$ are the Scheduler Differentiation Parameters and R is the link capacity. Since BPR is based on rate allocation, one can easily integrate link sharing policies and throughput differentiation into BPR. Unfortunately, BPR's performance in terms of proportional delay differentiation is remarkably worse than WTP when load distribution between classes of service is asymmetric [3].

In the next sections we will describe another scheduling mechanism, which we call *Differentiated Delay and Throughput Scheduler* (DDTS). Our goals are to develop a scheduling architecture that is able to perform delay and throughput differentiation and link sharing simultaneously. The scheduling policies are integrated in a single service discipline. Under different load patterns, the new scheduling architecture should keep its delay differentiation performance predictable and as close to the performance of WTP scheduler as possible, thus eliminate the shortcoming of the BPR scheduler.

3 Scheduling Algorithms

In this section, we firstly define a service architecture for the Differentiated Delay and Throughput Scheduler. The following part presents algorithms for two service differentiation models under DDTS, that is, delay and throughput.

The definition of DDTS is based on the concept of Packetized Generalized Processor Sharing [7] and the link-sharing model presented in [8,9]. A DDTS

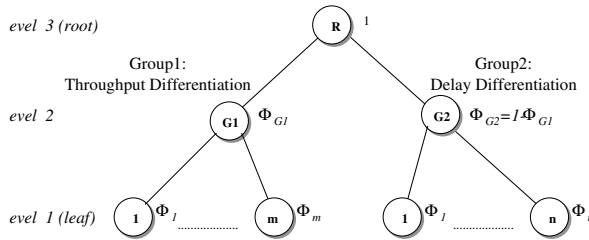


Fig. 1. Service architecture of DDTS

server can be logically presented by a three-level tree with a positive weight Φ_n ($\Phi_n < 1$) associated with each node n in the tree. The root node corresponds to the physical link with capacity R and each leaf node corresponds to a class of service with a queue (Fig. 1).

In the architecture, two *groups of service* are defined. The first group of service (G1) is for throughput differentiation and the second one (G2) is for delay differentiation. For the sake of simplicity, in DDTS the weight Φ_i assigned to a node i coincides with the percentage of total link capacity that node i will take up in case all sessions are back-logged, that is:

$$\Phi_{G1} + \Phi_{G2} = 1 . \quad (5)$$

$$\sum_{i \in G1} \Phi_i = \Phi_{G1} . \quad (6)$$

$$\sum_{j \in G2} \Phi_j = \Phi_{G2} . \quad (7)$$

where Φ_{G1} and Φ_{G2} are the link-shares of group 1 and group 2, respectively. Φ_i is the weight assigned to leaf node i and $B_{Gi}(t)$ is the set of all back-logged sessions of group i at time t . It is worth noting that other types of service, such as some classes of assured services, can easily be added in DDTS by using the link-sharing model in Figure 1. In that case, several level-2 nodes are added to present the link shares of the new classes. From the implementation point of view, the service architecture of DDTS is realized by a single scheduler, which is able to carry on different policies for different classes of service. DDTS comprises two components: the first one is a *work-conserving scheduler*, which can be implemented with any of the rate-proportional servers, e.g., WFQ [7,11], WF2Q [10] or SCFQ [12]. We assume that readers are already familiar to these rate-proportional algorithms. The second component is the *measurement and rate adaptation module*. Its functions are to measure the average delays of delay differentiation classes and to adjust their service rates so that delays between two adjacent classes of service are consistently spaced according to the Scheduler Differentiation Parameters (Eq. 3), independent of the current load pattern.

3.1 Scheduling Policy for Throughput Differentiation

It is not difficult to perform throughput differentiation in DDTS due to its rate-proportional nature. The throughput differentiation model is defined as below:

$$\frac{W_i(t)}{W_j(t)} = \frac{\Phi_i}{\Phi_j} \quad \forall i, j \in G1 . \quad (8)$$

where $W_i(t)$ denotes service offered to class i ($i \in G1$) during time interval $(t, t + \Delta t]$. Φ_i is the weight pre-assigned to class i , such that $\sum_{i \in G1} \Phi_i = \Phi_{G1}$. Here, Φ_i plays the roll of the quality differentiation parameter. Under such a rate-proportional scheduler as WFQ, WF2Q or SCFQ, Equation 8 holds for any back-logged session at time t .

3.2 Scheduling Policy for Delay Differentiation

Since packet delays are not directly related to rate-allocation in rate-proportional servers, it is more difficult to meet the requirements on delay differentiation. Delay spacing between classes can easily be deviated from the desired values as in case of BPR [3]. Let $s_i, i \in G2$ be the set of delay differentiation parameters. In the delay differentiation model, the average delays perceived by packets in any two delay classes are the inverse ratio of the corresponding delay differentiation parameters:

$$\frac{\bar{d}_i(t)}{\bar{d}_j(t)} = \frac{s_j}{s_i} \quad \forall i, j \in G2 . \quad (9)$$

The key concepts in DDTS are the introductions of the *equivalent queue length* $\tilde{q}_i(t)$ and the *adaptive differentiation parameter* $\sigma_i(t)$ for delay class i at time t . We rely on the idea of the BPR scheduler that the server allocates rates to the classes of service according to the current queue length of each session (see Equations 3 and 4), but instead of using real queue length, we calculate the equivalent queue length as follow:

$$\tilde{q}_i(t) = r_i(t)w_i(t) = R \frac{\Phi_i(t)}{\sum_{j \in \{B_{G1} \cap B_{G2}\}} \Phi_j(t)} w_i(t) \quad \forall i \in B_{G2} . \quad (10)$$

where $w_i(t)$ denotes the waiting time of the head-of-line packet of class i at time t , $r_i(t)$ denotes the rate allocated to class i at time t . The idea behind the equivalent queue length is that rate allocation in DDTS is directly associated with delay experienced by packets in the queue. Furthermore in some cases of non work conserving systems, queue length is not directly related to delay of packets. Allocating rate according to queue length in these cases might not be accurate.

The *measurement and rate adaptation module* measures average queueing delays in the node and uses these information as feedback signals to control the rates of delay classes so that delay spacing between classes of service is kept in a more precise manner, independent of different load patterns. Assume that packet

k from queue i leaves the system at time t , the *average normalized delay* over all delay classes and the *average delay of class i* are calculated as follows. We normalize delay $d_i(t)$ of packet k of class i with parameter s_i , then we calculate the average value of normalized delays over all delay classes. Let the *average delay* of class i be $\bar{d}_i(t)$, $A(x(t))$ be the average function of variable x at time t , we have:

$$\bar{d}_N(t) = A(s_i d_i(t)) . \quad (11)$$

$$\bar{d}_i(t) = A(d_i(t)) . \quad (12)$$

In DDTS, the server attempts to keep the normalized average delay of a class as close to $\bar{d}_N(t)$ as possible, that is, $s_i \bar{d}_i(t) \rightarrow \bar{d}_N(t)$. The *adaptive differentiation parameter* $\sigma_i(t)$ is defined as below:

$$\sigma_i(t) = \frac{s_i \bar{d}_i(t)}{\bar{d}_N(t)} s_i = \frac{s_i A(d_i(t))}{A(s_i d_i(t))} s_i = \frac{A(d_i(t))}{A(s_i d_i(t))} s_i^2 . \quad (13)$$

DDTS makes use of $\sigma_i(t)$ instead of s_i . Thus, rate allocation for delay differentiation classes in Equations 3 and 4 becomes:

$$\frac{r_i(t)}{r_j(t)} = \frac{\sigma_i(t)}{\sigma_j(t)} \frac{\tilde{q}_i(t)}{\tilde{q}_j(t)} \quad \forall i, j \in G2 . \quad (14)$$

$$\text{and} \quad \sum_{i=1}^N r_i(t) = R \frac{\Phi_{G2}}{\sum_{k \in \{B_{G1} \cap B_{G2}\}} \Phi_k(t)} . \quad (15)$$

From (13) and (14) it is remarked that, if $s_i \bar{d}_i(t) = \bar{d}_N(t)$ then $\sigma_i(t) = s_i$. Also, if $s_i \bar{d}_i(t) > \bar{d}_N(t)$ then $\sigma_i(t) > s_i$. Consequently, the average delay of class i tends to reduce in the next period. The deployment of the adaptive differentiation parameter $\sigma_i(t)$ in DDTS is a mechanism to provide consistent differentiation between classes, independent of varied load pattern. We will verify this in the next Section.

It is noted that different average functions $A(\cdot)$ can be used. In our approach, we use the *exponential window moving average* algorithm:

$$\bar{x}_k = \omega x_k + (1 - \omega) \bar{x}_{k-1} . \quad (16)$$

where x is a variable in time domain t , x_k and $\bar{x}_k$ are a sample of x and the exponential moving average at step k , respectively. ω is a weight with constraint $0 < \omega < 1$.

3.3 Complexity

The *measurement and rate adaptation module* in DDTS is responsible for (1) delay measurement, (2) weight calculation and (3) update of Finish Tags for head-of-line packets. These operations have to be done each time a packet leaves

the system. While DDTS measures delay only for the queue that has the departing packet, operations (2) and (3) need to be invoked for every active queue in the system. However, these operations can be efficiently implemented today, since each queue in DDTS is for traffic aggregate rather than for a single flow and the number of classes of service in DiffServ are normally limited to some tens. DDTS has the complexity of $O(n)$, where n is the number of classes of service.

4 Evaluation of DDTS Algorithms

In the last Section, algorithms for throughput and delay differentiation are shown. We will verify the algorithms through simulation experiments in this Section. Our objectives are to compare performance of DDTS and WTP in the context of delay differentiation under varied load patterns and link utilization, and to evaluate link sharing, throughput and delay performance in case of heterogeneous traffic. All simulations are implemented under *ns-2.1b5 network simulator* [13].

The first and second experiments compared the delay performance between DDTS and WTP. 40 Pareto sources are used to generate packets into a node with DDTS or WTP server. The server serves $N = 4$ queues, one for each delay differentiation class. The Pareto sources have the average *burst time* 0.35s, the average *idle time* 0.65s, *rate* (in “ON” state) 32 kbps and the *shape parameter* 1.9. We ran the simulations in 600 seconds and collected data from the last 400-second period, the first 200 seconds of the simulation are to bring up the system into steady state. The exponential moving average weight is set to $\omega = 0.05$ in all simulations.

The first experiment intended to test the performance of DDTS in terms of delay spacing between classes under different link utilization. The load distribution between delay differentiation classes is set to: Class-1: 40%, Class-2: 30%, Class-3: 20%, Class-4: 10% of the total load. There are 2 scenarios having been simulated: in the first scenario, the ratio s_i/s_{i-1} is equal to 2 and in the second one, it is equal to 4. In each scenario, we examined the performance of DDTS and WTP schedulers when the total traffic varies from moderate (75%) to heavy load (99%). From Figure 2, it is noteworthy that the average delay ratios of the both schedulers deviated remarkably from the desired values in moderate loads, while the proportional delay differentiation can be maintained more accurately in heavy-load situations.

The second experiment aimed to investigate the delay ratios between two successive classes under different load distributions. Similar to the first experiment, two scenarios with the ratio s_i/s_{i-1} equal to 2 and 4, respectively, have been simulated. Link utilization in the second experiment is fixed to 95% (Figure 3). There are 7 simulations in each scenario, in which the load pattern between four delay classes varied from symmetric to asymmetric distributions. The four numbers in each bar from Figure 3 denote the load distributions of the classes in percentage (legend of these bar graphs is similar to that in Figure 2).

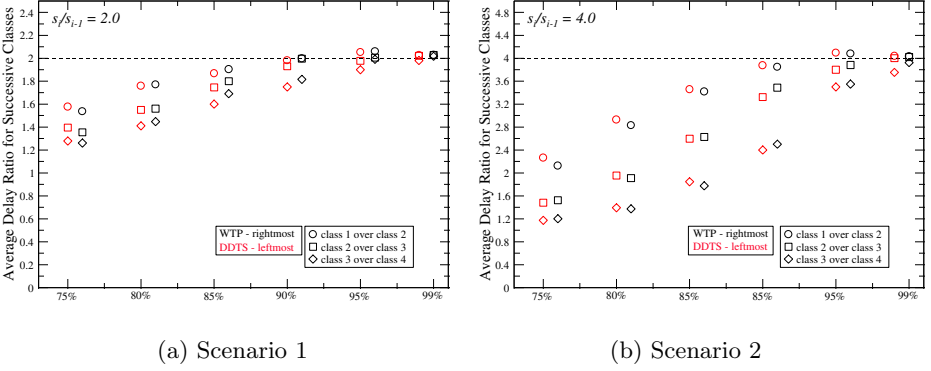


Fig. 2. The average delay ratios between successive classes under DDTS and WTP

Results derived from the both experiments showed that in most cases, the performance of DDTS is nearly the same the performance of WTP. This general trend is quite satisfactory.

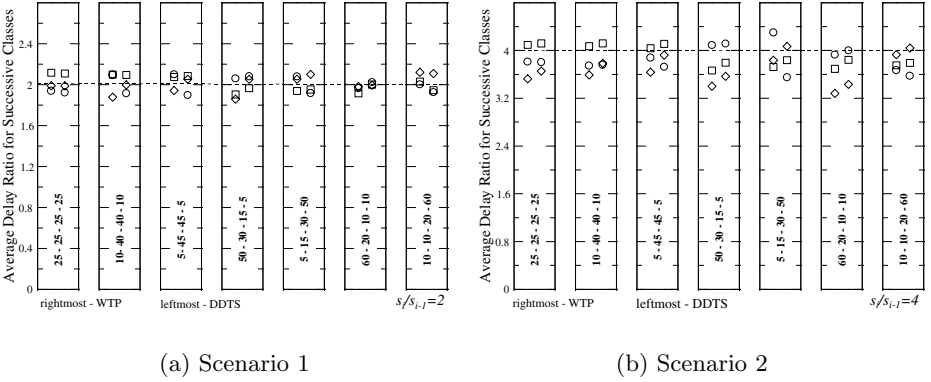


Fig. 3. The average delay ratios under different load patterns

In the last experiment, we investigated throughput and delay performance of DDTS in case multiple types of traffic present in the network. There are 8 classes of service in the simulated model: the first four classes are categorized into delay classes while the rest four classes belong to throughput differentiation classes. The link shares of delay and throughput differentiation classes are $\Phi_{G1} = 0.7$, $\Phi_{G2} = 0.3$, respectively. The output link capacity is set to $R = 4Mbps$, equivalent to 2.8Mbps for delay classes and 1.2Mbps for throughput classes. Real MPEG video sources and TCP sources are used in the simulation. Each video source consists of 40,000 MPEG frames (about half an hour)[14]. Class 1 consists of 4 video traces: 1 *Jurassic Park*, 1 *Starwars* and 2 *video conferencing* sessions; Class

2: one trace of *German news*; Class 3: 1 *Asterix*, 1 *German news*; Class 4: 3 *video conferencing* sessions. Each class from 5 to 8 has one FTP/TCP Tahoe source. The link utilization is about 98%. The sets of delay differentiation parameters $\{s_i\}$ and weights $\{\Phi_i\}$ are displayed in Figure 4.

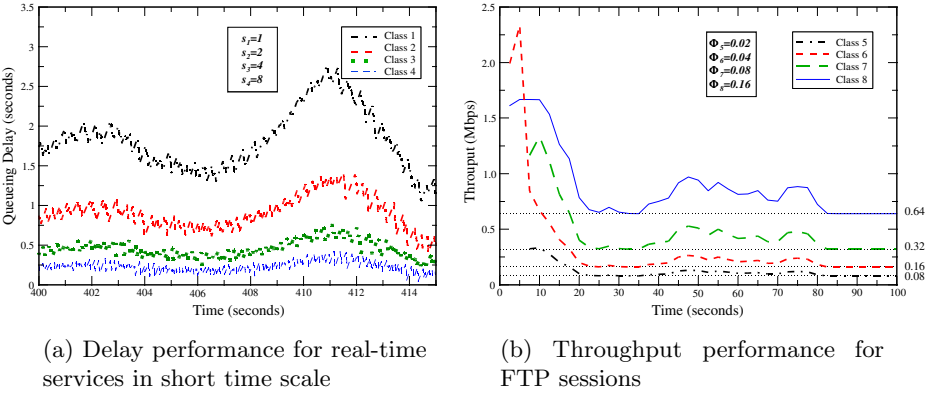


Fig. 4. Delay and throughput performance for different services

The start times at which the sources begin sending data into the network are randomly distributed. These random numbers are between the 0 and 10th second for FTP sources and between the 10th and 30th second for video connections. Delay differentiation in short time scale is maintained for real-time services as illustrated in Figure 4.a. For throughput differentiation sessions, as depicted in Figure 4.b, the FTP connections take advantage of free bandwidth to send more data into the network, when the video sources are idle. As other sources begin transmitting packets, their allocated bandwidths reduce gradually and converge into the nominated rates, independent of the window sizes set for each TCP source. The ratios between bandwidths allocated to successive classes are maintained independently of the load situation.

5 Conclusion

In the paper we developed a new scheduling mechanism called Differentiated Delay and Throughput Scheduler. A combined service model is proposed, in which delay differentiation services and link sharing mechanisms coexist. Delay differentiation in DDTS is as accurate as in WTP. Our measurement-based approach used in DDTS enables the network node to maintain service differentiation between classes of service, independent of load distribution between the classes. Furthermore, our approach supports an integration of delay and throughput differentiation classes and different link sharing strategies into a single scheduler. The advantages of the new scheduling mechanism are:

- *Simplicity*, in the sense that there is no need to deploy a multi-level scheduling architecture for multiple types of services and link sharing, thus it allows to reduce the implementation complexity at the network node.
- *Flexibility*, meaning that by use of DDTS, the network operator should be able to provide not only delay differentiation but also other services, such as bandwidth reservation for assured services, load control for best-effort services, link sharing or traffic engineering.

We like to emphasize that our work is still far from completion. The introduction of throughput differentiation classes in the article is an example for rate-proportional allocation and link sharing rather than a new class of service, since higher throughput allocated to a class does not mean that a single flow belonging to that class will receive more bandwidth than flows belonging to “worse” classes. Here, it might require that users deploy an adaptation mechanism to dynamically adjust the flow’s class in order to achieve an appropriate quality of service, which is subjected to future research.

References

1. R. Braden, D. Clark, S. Shenker: Integrated Services in the Internet Architecture: an Overview. RFC 1633, IETF (1994)
2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss: An Architecture for Differentiated Services. RFC 2475, IETF (1998)
3. C. Dovrolis, D. Stiliadis, P. Ramanathan: Proportional Differentiated Services: Delay Differentiation and Packet Scheduling. In Proceedings of SIGCOMM 1999
4. C. Dovrolis, P. Ramanathan: Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping. In Proceedings of IWQoS 2000
5. Stefan Bodamer: A Scheduling Algorithm for Relative Delay Differentiation. In Proceedings of ATM 2000
6. T. Nandagopal, N. Venkitaraman, R. Sivakumar, V. Bharghavan: Delay Differentiation and Adaptation in Core Stateless Networks. In Proceedings of INFOCOM 2000
7. Abhay K. Parekh, Robert G. Gallager: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case. IEEE/ACM Transactions on Networking, Vol. 1, No. 3 (1993)
8. Sally Floyd, Van Jacobson: Link-sharing and Resource Management Models for Packet Networks. IEEE/ACM Transactions on Networking, Vol. 3, No. 4 (1995)
9. Jon C. R. Bennett, Hui Zhang: Hierarchical Packet Fair Queueing Algorithms. IEEE/ACM Transactions on Networking (1997)
10. Jon C. R. Bennett, Hui Zhang: WF2Q: Worst-case Fair Weighted Fair Queueing. In Proceedings of INFOCOM 1996
11. A. Demers, S. Keshav: Analysis and Simulation of a Fair Queueing Algorithm. Journal of Internetworking Research and Experience (1995)
12. S. Golestani: A Self-clocked Fair Queueing Scheme for Broadband Applications. In Proceedings of INFOCOM 1994
13. The Network Simulator – ns-2, <http://www.isi.edu/nsnam/ns/>
14. O. Rose: Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems. In Proceedings of 20th Annual Conference on Local Computer Network 1995.

Evaluation of an Algorithm for Dynamic Resource Distribution in a Differentiated Services Network

E.G. Nikolouzou, P.D. Sampatakos, and I.S. Venieris

National Technical University of Athens
School of Electrical and Computer Engineering
Telecommunications Laboratory
9 Heroon Polytechniou str, 15773 Athens, Greece
{enik, psampa}@telecom.ntua.gr, ivenieri@cc.ece.ntua.gr

Abstract. New applications have been introduced to the today's "best-effort" IP networks having different bandwidth and delay guarantee requirements. The IETF is currently focused on Differentiated Services as the architecture to provide Quality of Service to IP networks. Towards this effort, an overlay Resource Control Layer on top of a Differentiated Services core network is introduced in this paper, in order to provide a simple control plane architecture that enables the overall handling of network resources and the configuration of network elements in a domain. Therefore, a dynamic algorithm is proposed for that layer to manage, adjust and distribute resources in an efficient and dynamical way. The simulation results show that this algorithm provides a significant improvement in bandwidth assurance and utilization of network resources compared with a static resource assignment approach, keeping at the same time complexity at a low level.

Introduction

The Internet today provides a best-effort architecture, which is basically ideal for elastic applications, such as e-mail and file transfer. The network traffic though has increased as the number of users and applications has also increased. Moreover, the Internet traffic has also changed in character; new bandwidth-demanding and delay-sensitive applications (voice-over-IP, IP-telephony, video-conferencing) require or at least benefit from Quality of Service (QoS) [1,2] or other form of prioritisation that guarantees an Internet connection. Increasing bandwidth is not always sufficient to accommodate these increased demands. QoS mechanisms provide expected and predefined service guarantees by better managing the available bandwidth.

The Differentiated Services (DiffServ) architecture [3,4,5,6] is nowadays the preferred architecture, which can address quality of service issues in IP networks. It provides a coarse and simple way to categorize and prioritize network traffic (flow) aggregates, leaving complexity at the "edges" and keeping the "core" network simple enabling its scalability. Edge devices (ED) in this architecture perform packet classification, policing, shaping and marking in order to ensure that individual user's traffic conforms to the specified traffic profiles and aggregate traffic into a small number of prioritized classes. Core routers treat packet aggregates with Per-Hop-

Behavior (PHB) [7,8] according to their markings. PHB is the forwarding treatment that a packet receives at a network node. The concept of the Bandwidth Broker (BB) Architecture [9,10] was proposed by Internet2 in order to provide an overall resource management, policy-based admission control and configuration of specific network elements (leaf, core and border routers).

Our proposed architecture is based on the DiffServ and BB [11,12] concept. It is basically a realization of a distributed BB architecture, promising scalability and efficiency. Consequently, an additional layer on “top” of the DiffServ architecture is realized, the Resource Control Layer (RCL) as described in the [13]. The RCL is composed of different distributed entities each one assigned a specific task. The algorithm realized and evaluated in this paper is responsible for the resource management performed by the RCL. The rest of the paper is structured as follows: in the following part an outline of the proposed architecture is presented. In the last section, the implemented algorithm is shortly described and evaluated.

Motivation & Proposed Architecture

The architecture proposed aims at an efficient management and distribution of resources between the different nodes of a DiffServ architecture. This is basically realized by the proposed algorithm implemented in this layer, which achieves a good utilization of network resources. The architecture is fully analyzed in [13], and here its main functionality is described. It is composed of three logical entities. To start with, the Resource Control Agent (RCA) is the highest control entity in an administrative domain and is responsible for configuring the appropriate network entities and managing the network resources. Moreover, it has the overall view of the policies enforced in a domain and decides for the management of bilateral Service Level Agreements between adjacent administrative domains. Second, the Admission Control Agent (ACA) performs admission control based on the traffic profile between the user and the network. In this way, it controls the access of the user to the network and performs authorization and usage metering (accounting) functions. Last, the End-User Application Toolkit (EAT) provides a graphical interface to end-user applications and enables them to signal their requirements to the QoS infrastructure. The above logical entities can be distinguished in Fig. 1.

In order the RCA entity to manage more efficiently the resources distributed among the networks elements, a hierarchical architecture inside the RCA is proposed. Therefore, instead of having a centralized resource management entity, a distributed one is proposed, separating the network to sub-networks. Each sub-area has its own initial resources, which are assigned according to traffic loads forecasts and/or results retrieved by a measurement-based platform. The structure of the RCA is depicted in Fig. 2.

The resources assigned to the administrative domain (root) are distributed among the sub-areas, each one represented by a Resource Pool (RP). Moreover, each sub-area can also be further divided into sub-areas, forming the above hierarchical structure. Another reason for the creation of RPs is the correct management of bottleneck links and the efficient sharing of its bandwidth between the RPs of the lower level. The Resource Pool Leafs (RPLs) correspond to the resources assigned to each ACA. Each ACA is based on those resources to perform admission control. The

assignment of the resources is a top-down procedure, from the root of the tree down to the RPLs. On the left hand of the Fig. 2. is given an example of RPs creation based on the network of Fig. 1. and on the right hand a more complicated hierarchical structure.

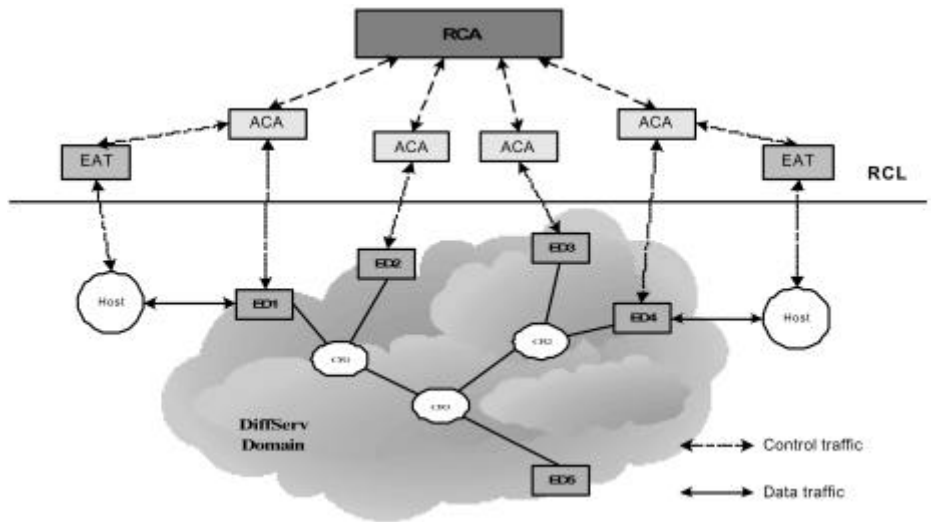


Fig. 1.: RCL infrastructure

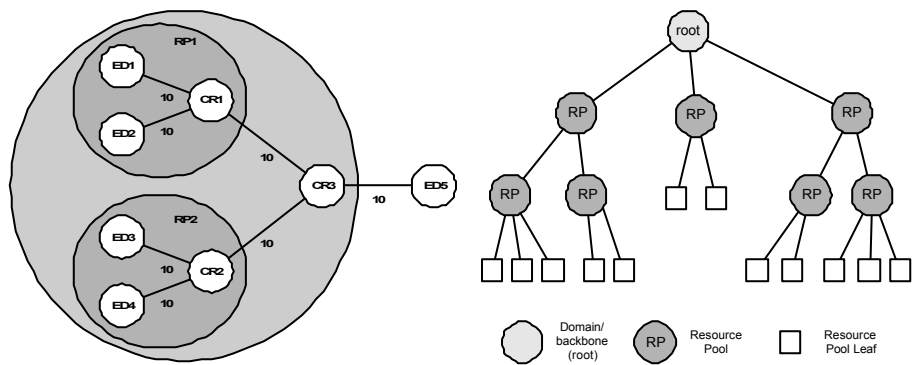


Fig. 2.: Hierarchical structure of RCA

The initial assigned resources may not correspond to actual traffic load, therefore, the RPLs/RPs are capable of adjusting and adapting those initial resource assignments to real traffic conditions, which are difficult to be forecasted and may change during time.

The Algorithm

The main target of the algorithm is to efficiently handle the re-distribution of resources. This is invoked when an RPL does not have enough resources to accommodate a new user request. Each RP and RPL is basically described by the following set of parameters:

- R_{max} : upper limit of resources that can be assigned to an RP/RPL.
- R_{tot} : current resource assignment to an RP/RPL
- R_{res} : current reserved resources of an RP/RPL
- R_{free} : currently free resources of an RP/RPL
- R_{add} : maximum resources that can be additionally assigned to an RP/RPL

The equations (1)-(6) describe the initial resource status of an RP/RPL as well as the relation of the resources of a father RP and its children (f: father, c: children):

$$R_{max} \geq R_{tot} \geq 0 \quad (1)$$

$$R_{free} = R_{tot} - R_{res} \quad (2)$$

$$R_{add} = R_{max} - R_{tot} \quad (3)$$

$$R_{res}^f = \sum R_{tot}^c \quad (4)$$

$$R_{max}^f \geq R_{max}^c \quad (5)$$

$$\sum R_{max}^c \geq R_{max}^f \quad (6)$$

The network administrator is responsible for defining the initial resources to be distributed to the nodes of the tree. After this top-down start-up procedure, initial resources are assigned to all nodes of the tree. Sequentially a user can make its resource reservation requests to the EAT, which forwards these requests to the ACA. Under the condition that the user access to the network is verified, ACA hands over this request to the corresponding RPL for admission control.

According to the algorithm realized, an RPL will make a request for additional resources to its father when its current free resources are not adequate to serve a new request. The child makes a request and the father is responsible for deciding how many resources to give to its child, depending on the amount of resources requested, the upper limit defined by the child (R_{add}) and the amount of its free resources. In case the father does not have enough resources will also make a resource request to its father RP (of the above level). This procedure can continue up to the root of the tree. The procedure of finding additional resources is bottom-up, i.e. from the leaves of the tree up to the root.

A number of additional parameters must be defined for the realization of the algorithm:

- R_{req} : minimum resources requested from an RP/RPL
- R_{recv} : resources actually received from a child after a request for more resources to its father
- A_{max} : number of max resource shifts; father RP increases the resources of its child by $A_{max} \times R_{req}$
- A_{med} : number of med resources shifts; father RP increases the resources of its child by $A_{med} \times R_{req}$
- A_{min} : number of min resource shifts; father RP increases the resources of its child by $A_{min} \times R_{req}$ ($A_{max} > A_{med} > A_{min} \geq 1$)
- ρ_L : a low limit for the free resources of the RP, $\rho_L < 1$
- ρ_H : a high limit for the free resources of the RP, $\rho_H > 1$ ($\rho_H > \rho_L$)

The ρ_L and the ρ_H determine two limits for the free resources of an RP. Actually a low and a high watermark are defined corresponding to $\rho_L \times R_{tot}$ and $\rho_H \times R_{tot}$.

As long as the RPL has enough resources to accept a reservation request, there is no need of redistribution of the resources. In case an RPL does not have efficient resources to accommodate an R_{req} it asks more resources from its father RP, and the latter decides how much to give back to it, R_{recv} . The same procedure can be repeated many times, up to the root of the tree. The steps of the proposed algorithm executed by the RPL after a resource reservation request are:

1. **if** $R_{res}^{RPL} + R_{req} > R_{tot}^{RPL}$ **then reject the request;**
2. **if** $R_{res}^{RPL} + R_{req} \bullet R_{tot}^{RPL}$ **then admit request** $R_{res}^{RPL} = R_{res}^{RPL} + R_{req}$
end then (2)
3. **else if** $R_{res}^{RPL} + R_{req} > R_{tot}^{RPL}$
then calculate resources to ask from father $R_{ask} = (R_{res}^{RPL} + R_{req}) - R_{tot}^{RPL}$ **make**
a request to father $R_{recv} = request(R_{ask}, R_{add}^{RPL});$
if request accepted by father RP then admit the request change total and
reserved resources:
 $R_{tot}^{RPL} = R_{tot}^{RPL} + R_{recv}, R_{res}^{RPL} = R_{res}^{RPL} + R_{req}$ **end then**
else reject the request; end then(3)

In case a father RP can not assign to its child not even the minimum amount of resources requested, it requests in the same way resources from its corresponding father. The father RP uses the following algorithm in order to calculate the resources to give back to its child. The father RP basically compares its low and high watermark of free resources with a multiple of the resources requested. Depending on the result of the comparison, it gives back an appropriate multiple ($A_{max}/A_{min}/A_{med}$) of the resources requested.

1. **if** $A_{max} \times R_{ask} < \rho_L \times R_{free}$ **then** $R_{recv} = \min(A_{max} \times R_{ask}, R_{add}^c), R_{res} = R_{res} + R_{recv}$
return R_{recv} **end then (1)**
2. **else if** $A_{med} \times R_{ask} < \rho_H \times R_{free}$ **then** $R_{recv} = \min(A_{med} \times R_{ask}, R_{add}^c), R_{res} = R_{res} + R_{recv}$
return R_{recv} **end then (2)**
3. **else if** $A_{min} \times R_{ask} < R_{free}$ **then** $R_{recv} = \min(A_{min} \times R_{ask}, R_{add}^c), R_{res} = R_{res} + R_{recv}$
return R_{recv} **end then (3)**
4. **else ask resources from its father**
 $R'_{ask} = (R_{res} + R_{ask}) - R_{tot}$
 $R_{recv} = request(R'_{ask}, R_{add})$
if request accepted by father then $R_{tot} = R_{tot} + R'_{recv},$ **goto step(1)** **end then**
else reject the request; end then (4);

When the ACA makes a release request to the RPL, the latter de-allocates the corresponding resources and checks whether or not it can give back any free resources to its father. In order to take such decision an additional set of variables are defined:

- l : a low limit of the R_{tot} , $l < 1$
- R_{rel} : requested resources to be released
- R'_{rel} : resources to be given back to the upper level
- a : it determines indirectly the actual amount of resources to be returned, $a < 1$

The low watermark, $l \times R_{tot}$, is used to check the current status of reserved resources of an RP/RPL. In case the reserved resources before the release are above the low watermark and the resources after the release are below this watermark, then an amount of free resources should be returned to the upper level. The purpose of this double check of resources is to control that an RP/RPL is not actually in an initial

state, where resource reservations have just began. In that case its reserved resources may not have yet exceeded the low watermark so that resources should not be returned to the upper level. The amount of resources to be given back should be calculated considering the trade-off between giving as much as possible and keeping resources for future use. This calculation is actually based on the desired level of reserved resources between the total resources and the low watermark. The value of a determines this level.

The algorithm for deciding and calculating the resources to be returned is:

1. *After the release:* $R'_{res} = R_{res} - R_{rel}$
2. **if** ($R'_{res} < l \times R_{tot}$) **and** ($R_{res} > R_L$)
then have to give back resources to the upper level so that reserved resources to be between the R'_{tot} and $l \times R'_{tot}$:
 $R'_{res} = a (R'_{tot} + l \times R'_{tot})$, where $R'_{tot} = R_{tot} - R'_{rel}$
From above: $R'_{rel} = R_{tot} - R_{res} / (a (1 + l))$ **else** do not give back resources
end then (2)

Simulation

Simulations were carried out in a Pentium III PC with the help of a special tool that has been developed in JAVA programming language. In order to understand fully the behavior of the algorithm, a tree structure has been defined and implemented, as depicted in

. The actual tree structure does not play a crucial role for the study of the proposed algorithm.

A simulation experiment consists of a random process of reservation request arrivals. Each request arriving to an RPL may be admitted or rejected according to the specifics of the algorithm in question. The inter-arrival time of reservation requests follows an exponential model, while the size of the resources requested have a standard capacity of 128kbps. Each leaf node has a weight, which determines the amount of initial resources assigned to it. Those initial resources in a real network could have been based on some load forecasts. The offered load to the leaf nodes differs from the one forecasted in order to prove the adaptability of the algorithm. While the resources are distributed to nodes 1,2,3 with weights 0.5, 0.3, 0.2, the actual offered load is correspondingly 0.5, 0.4, 0.1 for the half time of simulation time and 0.5, 0.2, 0.3 for the rest time.

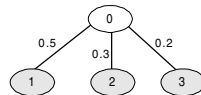


Fig. 3: Simulation topology

In order to verify the performance achieved by the proposed algorithm, it is actually compared to a static configuration, where the concept of resource pools is not used. An amount of resources is assigned to each ACA, which do not change during simulation. Moreover the behavior of the proposed algorithm has been examined

under different set of values of parameters. Table 1. summarizes those parameters and assigns to them a possible value.

Table 1. Main variables of the algorithm

Variable	Value
A_{max}	5(3-8)
A_{med}	3(2-4)
A_{min}	1(1-3)
w_L	0.2(0.2-0.5)
w_H	0.6(0.5-0.8)
L	0.5(0.5-0.7)
A	0.5

Primarily we have examined the variation of R_{tot} and R_{res} in time for all the RPs/RPLs of the tree, changing the values of the parameters in Table 1. . In general the algorithm offers an exceptional adaptability as indicated in Fig. 4 for an RPL and A_{max} is set to the value of 7. The adaptability of R_{tot} to the reserved resources, R_{res} , depends mainly on the values of A_{max} and l . The greater the value of A_{max} the less adaptive the algorithm becomes, since a greater amount of resources will be re-assigned to a child after a request() call. The value of l determines the level that resource release must be, meaning that the greater its value is, the sooner unused resources will be released to the upper level.

In sequence the number of interactions among all nodes of the tree was examined for different values of A_{max} . As a result of the simulations the greater the value of A_{max} the smaller the number of interactions.

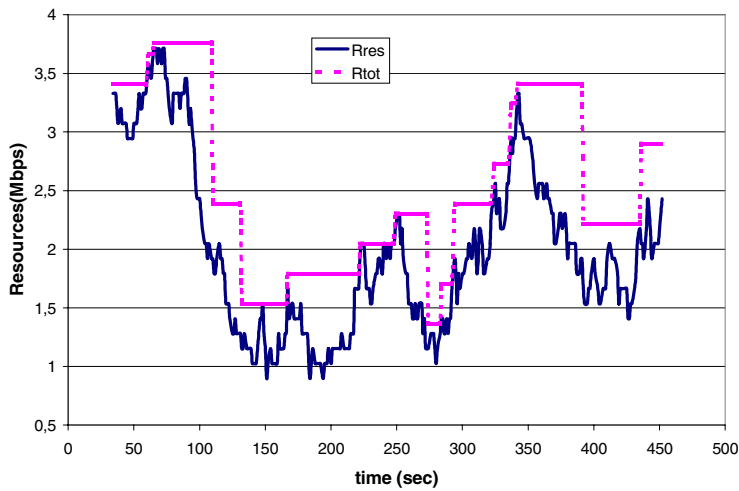


Fig. 4: Status of Resources of a RPL

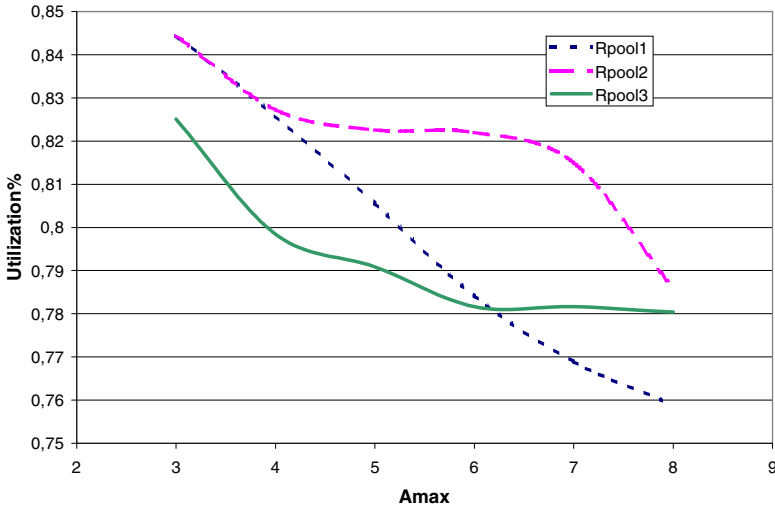


Fig. 5: Utilization in relation to A_{max}

Another crucial characteristic for the performance of the proposed algorithm is the utilization of the network resources. The average utilization has been measured for each leaf varying the value of A_{max} from 3 to 8, as illustrated in Fig. 5. The algorithm really provides a high utilization, which is inversely proportional to the value of A_{max} . The current utilization of resources of each node depends also directly on the value of l , since l composes an under bound for the utilization.

It has been also examined the response of the algorithm to the modification of values of the other parameters. A_{min} , A_{med} , w_H and w_L also influence the utilization and the number of interactions in the same way as A_{max} , but they have a smaller impact than A_{max} . In addition the behavior of the parameter a is identical to that of l , since they both determine the state that release of resources should take place.

Finally the number of rejected resource requests has been measured for the proposed as well as the static algorithm, as depicted in Fig. 6. The nodes 1 and 3 under the proposed algorithm invoke no rejections while node 2 (RPL2) generates a small number of rejections. The nodes under the static algorithm generate a number of rejections, which are proportional to the offered load. It is really obvious how the proposed algorithm outperforms the static version, offering a really smaller number of rejections, since it achieves a dynamic resource distribution between the leaves of the tree.

Summarizing, there is trade-off between the utilization of network resources and the interactions between the nodes of the tree. When the main goal of the implementation is a small number of interactions among the remote nodes for improving the performance, then a relatively large value of A_{max} is required. Consequently, a smaller utilization of network resources is achieved. It depends also on the network administrator to tune appropriately the value of A_{max} and the other parameters in order to achieve the desired performance. In addition it has been verified the significant improvement in bandwidth assurance and resources utilization

of the proposed algorithm compared to a static version, which keep though the complexity at a really low level.

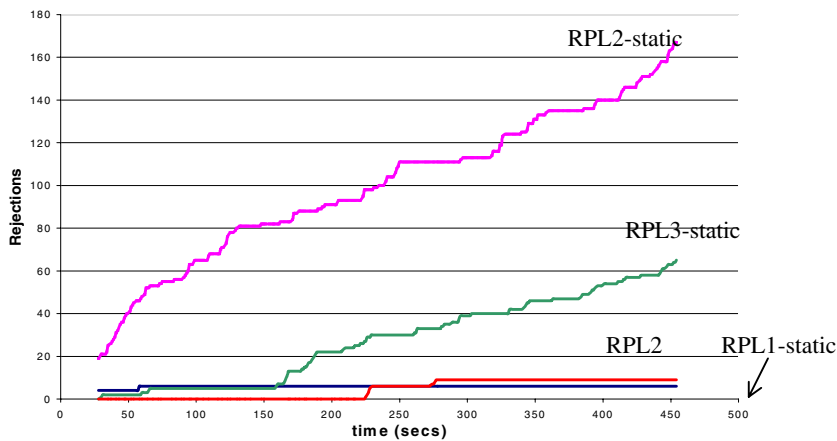


Fig. 6: Number of rejected resource requests

Conclusions & Future Work

The proposed realized algorithm uses some techniques in order to adapt efficiently and dynamically the resources of an RP/RPL to real traffic loads. The simulation results prove how this algorithm outperforms a static configuration, without a significant complexity burden.

A management platform is under study in order to provide a graphical interface for the monitoring and configuration of the RPs. In addition new versions of the proposed algorithm are planned for the future in order to further examine the role of the different parameters as well as to tune their value more properly.

Acknowledgements

This paper is partly funded by the European research project AQUILA, Information Societies Technology (IST) programme, IST-1999-10077. The authors are currently engaged in the definition, implementation and evaluation of the concept presented in this paper.

References

- [1] Ben Teitelbaum et al, "Qbone Architecture (v1.0)", Internet2 QoS Working Group Draft, August 1999, Work-in-Progress.
- [2] Ben Teitelbaum, Ted Hanss, "QoS Requirements for Internet2", April 22, 1998.
- [3] Black, D.; Blake, S.; Carlson, M.; Davies, E.; Wang, Z.; Weiss, W., "An Architecture for Differentiated Services", RFC 2475, 1998.
- [4] Y. Bernet, J. Binder, S. Blake, M. Carlson, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, W. Weiss, "A Framework for Differentiated Services", Internet Draft, October 1998.
- [5] Ben Teitelbaum, "Qbone Architecture (v1.0)", Internet2 QoS Working Group Draft, August, 1999, <http://www.internet2.edu/qos/wg/papers/qbArch/1.0/draft-i2-qbone-arch-1.0.html>.
- [6] D. Grossman, "New Terminology for Diffserv", Internet-Draft, November 1999.
- [7] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", IETF Proposed Standard RFC 2598, June 1999.
- [8] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.
- [9] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", RFC 2638, July 1999.
- [10] R. Neilson, J. Wheeler, F. Reichmeyer, S. Hares, "A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment", Internet2 Qbone BB advisory Council, August 1999.
- [11] J. Ogawa, Y. Nomura, "A Simple Resource Management Architecture for Differentiated Services", Internet Society, Japan, July 2000.
- [12] British Columbia Institute of technology, "CA*net II Differentiated Services-bandwidth Broker High-Level Design", November 1998.
- [13] G. Politis, P. Sampatakos, I.S. Venieris, "Design of a multi-layer bandwidth broker architecture", Lecture Notes in Computer Science; Vol 1938, Springer Verlag, Oct 2000.

Quality of Service Management in GPRS Networks

Peter Stuckmann and Frank Müller

Communication Networks
Aachen University of Technology
`pst@comnets.rwth-aachen.de`
<http://www.comnets.rwth-aachen.de/~pst>

Abstract. In this paper the performance and capacity gain achievable with quality of service (QoS) management in packet switched radio networks based on the General Packet Radio Service (GPRS) are examined. Both the functions defined in the GPRS specification for QoS support and implementation-specific strategies for subscriber- and application-based Connection Admission Control (CAC) and scheduling are introduced. Taking characteristic measures for a pure best-effort service as the basis the effect of these QoS functions on the throughput and delay is analyzed. To achieve this, simulation results of GPRS performance and system measures for different load situations are produced with the simulation tool GPRSim that models the realistic traffic behaviour of a GPRS network.

1 Introduction

In the context of the evolution towards 3rd Generation (3G) mobile radio networks, packet switched data services like the General Packet Radio Service (GPRS) and the Enhanced GPRS (EGPRS) are presently introduced into GSM and TDMA/136 systems worldwide [1,2]. While in the first phase only best-effort data services without differentiating subscribers and applications will be supported, in the second phase quality of service (QoS) management functions will be integrated to be able to guarantee subscriber- and application-specific QoS requirements. For radio network dimensioning and network equipment further development the effect of these QoS management functions on the overall system performance has to be determined. This paper does not aim at optimized solutions for QoS management functions, but at simple proposals to be able to estimate the performance gain achieved with their introduction compared to a pure best-effort service. The focus lies on the radio network and not on the core network, since radio resources are scarce and representing the system bottleneck assuming that the core network is well dimensioned.

In the next section the QoS functions in GPRS networks focussing on the radio network are introduced. Next the simulation model GPRSim comprising the implementation of QoS support is presented. This is the basis for the simulation results given in the next section that figure out the performance gain through QoS functions in GPRS compared to simulations for a pure best-effort service.

2 Quality of Service in GPRS Radio Networks

From a network operator's point of view, the introduction of QoS-based traffic management offers various advantages. Not only it is possible to utilize network resources in a more efficient manner by treating application data flows with respect to their actual needs – e.g., fetching an e-mail does not pose the same strict requirements on the packet delay as an IP telephone call – but also to differentiate between service users with respect to their subscribed QoS. Without an efficient QoS management upcoming real-time applications like VoIP or Video Conferencing will not even be possible to be supported. 3GPP has specified QoS requirements for different service classes namely Conversational, Streaming, Interactive and Background. Table 1 gives an overview of these requirements [3].

In this paper only the problem of QoS provisioning in the radio network is considered. The serving host is assumed to be located in the operator's domain and the core network is assumed to be well dimensioned.

To define a QoS contract between the mobile station (MS) and the network, Packet Data Protocol (PDP) contexts containing QoS profiles are negotiated between the MS and the Serving GPRS Support Node (SGSN) [4]. The Base Station Subsystem (BSS) is provided with a Packet Flow Context (PFC) containing the Aggregate BSS QoS Profile (ABQP) and is responsible for resource allocation on a Temporary Block Flow (TBF) base and scheduling of packet data traffic with respect to the according QoS profiles negotiated. Moreover, it regularly informs the SGSN about the current load conditions in the radio cell. The tasks of the Gateway GPRS Support Node (GGSN) comprise mapping of PDP addresses as well as classification of incoming traffic from external networks regarding the downlink Traffic Flow Template (TFT). The GPRS Register (GR) holds the QoS-related subscriber information and delivers it on demand to the GSN. In Figure 1 a GPRS session is schematically outlined depicting the instances involved, messages exchanged, and parameters used for PDP context (re)negotiation, PFC setup, and TFT installation.

From a time-scale point of view, the mechanisms for QoS management within the GPRS can be regarded as a three-stage model (see Figure 2). On PDP context activation the QoS parameters are negotiated. As long as the PDP context remains active, these parameters should be guaranteed unless there is a QoS renegotiation. The QoS profile is considered both for each TBF and for each

Table 1. QoS requirements for selected services belonging to different traffic classes

Traffic class	Medium	Application	Data rate	One-way delay
Conversational	Audio	Telephony	4–25 kbit/s	< 150 ms
	Data	Telnet	< 8 kbit/s	< 250 ms
Streaming	Audio	Streaming audio (HQ)	32–128 kbit/s	< 10 s
	Video	One-way	32–384 kbit/s	< 10 s
	Data	FTP	–	< 10 s
Interactive	Audio	Voice messaging	4–13 kbit/s	< 1 s
	Data	Web-browsing (HTML)	–	< 4 s/page

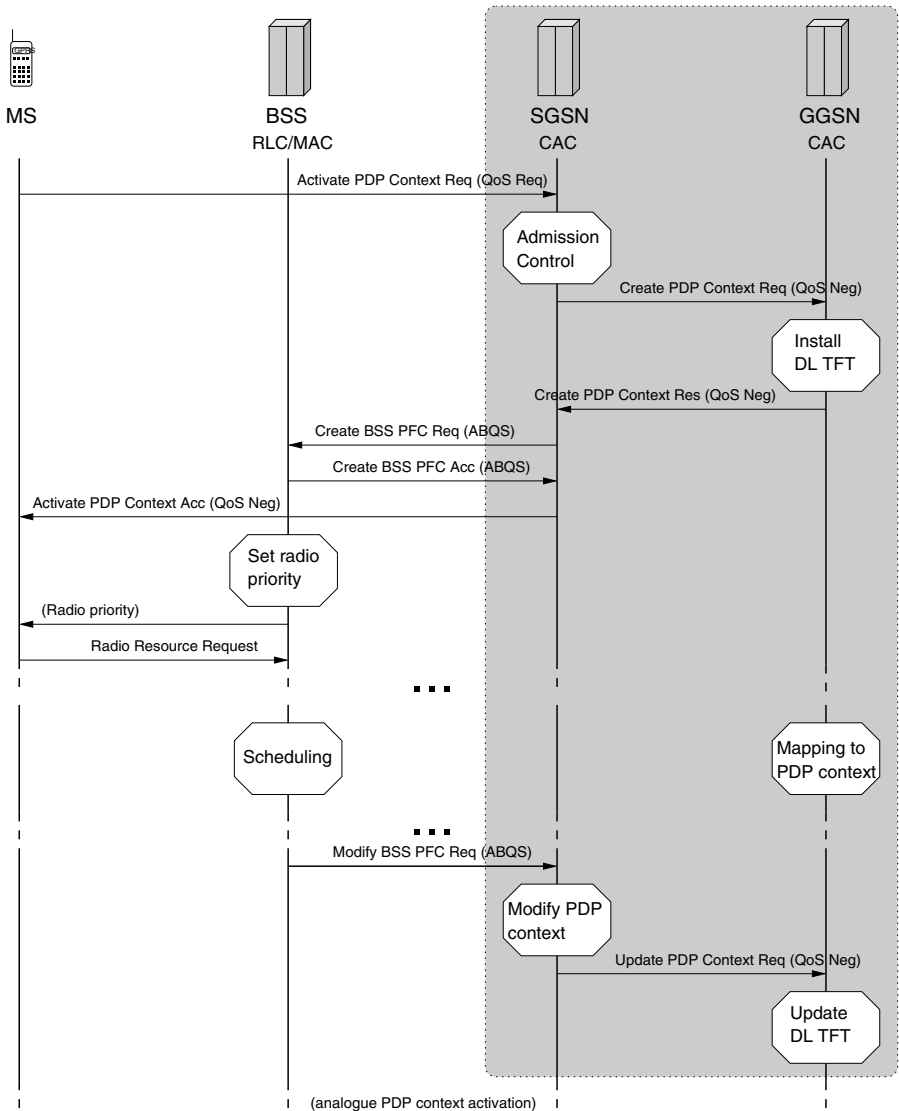


Fig. 1. QoS negotiation and renegotiation procedures (example)

radio block period. At TBF setup, radio resources like a set of Packet Data Channels (PDCH) usable for this TBF are assigned according to the negotiated QoS parameters. During the TBF, radio blocks are scheduled at the BSS in competition with other existent TBFs in the radio cell. This scheduling has to be done considering the QoS profiles of the PDP contexts associated with the TBF.

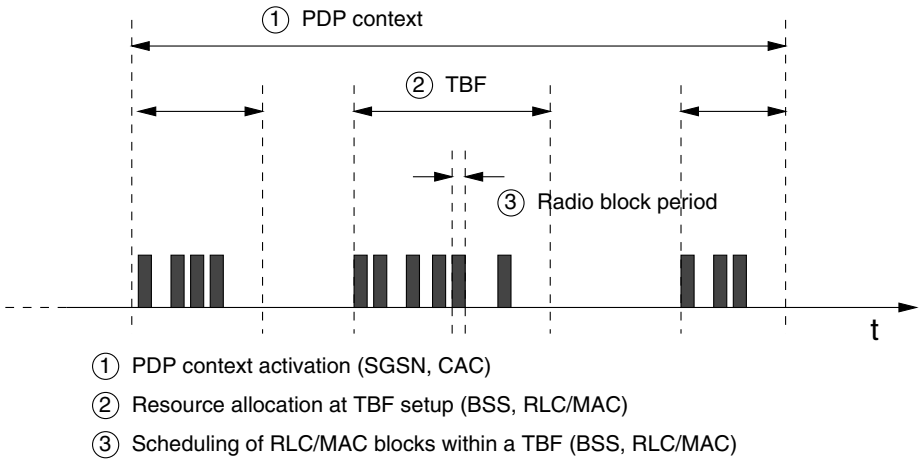


Fig. 2. Three-stage model of QoS management

3 Simulation

In this section the simulation tool GPRSim that is the basis for the performance analysis is introduced. This simulation environment allows to ascertain and to optimize properties of different protocols of the (E)GPRS transmission plane. In addition, it gives the possibility of network capacity and quality of service planning by performance evaluation in certain simulation scenarios. The GPRS/EGPRS Simulator GPRSim is developed as a pure software solution in the programming language C++. Models of Mobile Station (MS), Base Station (BS), and Serving GPRS Support Node (SGSN) are implemented. The simulator offers interfaces to be upgraded by additional modules (see Figure 3). For implementation of the simulation model in C++ the Communication Networks Class Library (CNCL) is used, which was developed at the Chair of Communication Networks. It allows an object oriented structure of programs and is especially applicable for event driven simulations. The complex protocols like LLC, RLC/MAC based on GPRS/EGPRS Release 99 and the Internet Load Generators including TCP/IP and UDP/IP are specified with the Specification and Description Language (SDL), translated to C++ by the Code Generator SDL2CNCL and finally integrated into the simulator.

Different from usual approaches to building a simulator, where abstractions of functions and protocols are being implemented, the approach of the GPRSim is based on the detailed implementation of the standardized protocols. This enables a realistic study of an (E)GPRS network.

The functions not specified in detail in the GPRS specification are the CAC policy and the scheduling strategy. The implementation of these components in the GPRSim is depicted in the following.

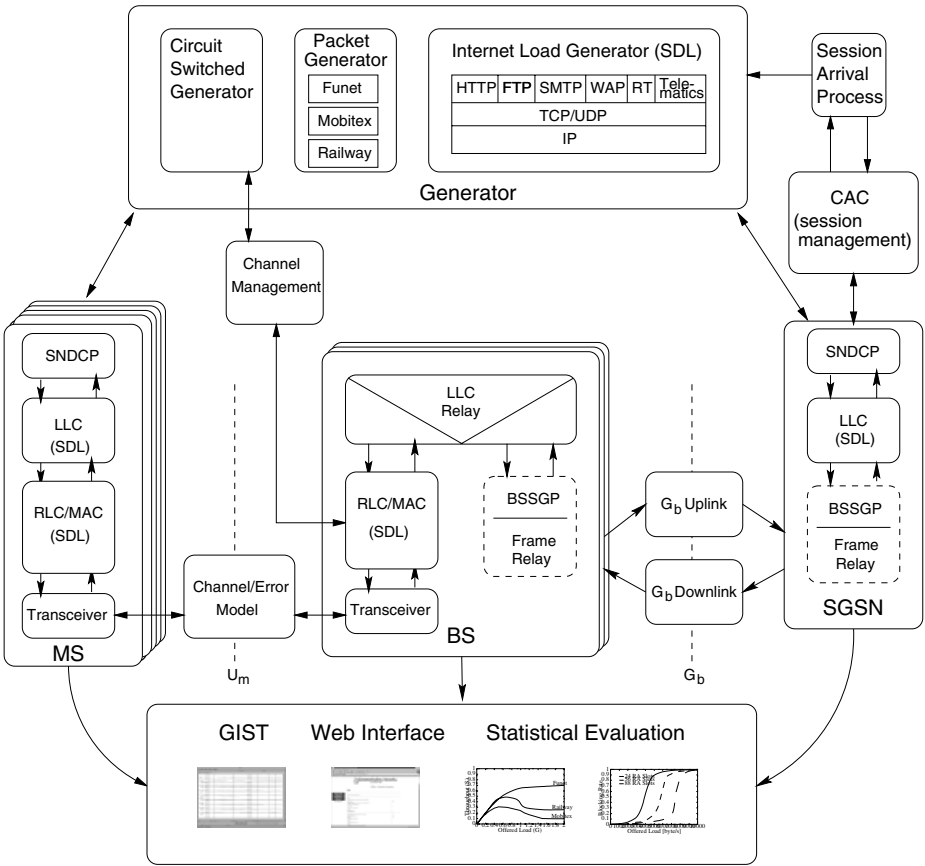


Fig. 3. The GPRS/EGPRS Simulator GPRSim

3.1 Connection Admission Control

In the simulation model PDP requests are differentiated on subscriber base (Premium, Standard, Best-Effort (BE)) and application base (Conversational, Streaming, Interactive, Background). In this study only Interactive (WWW) and Background (e-mail) are regarded, since these are the applications predicted for GPRS in the next years. To avoid a total withdrawal of resources from the Standard traffic classes with lower QoS requirements, e.g., other than Conversational, there is a share reserved for this kind of traffic from the pool of radio resources in the cell. In general, all resources are open to traffic of any kind. In time of high load, however, traffic flows with more demanding QoS requirements are allowed to displace flows belonging to applications with lower QoS requirements, but only up to a certain limit (see Figure 4), where P and I represent the appropriate limits. When this limit is reached, the requested QoS is not granted, but rather degraded to the next-lower-prioritized class.

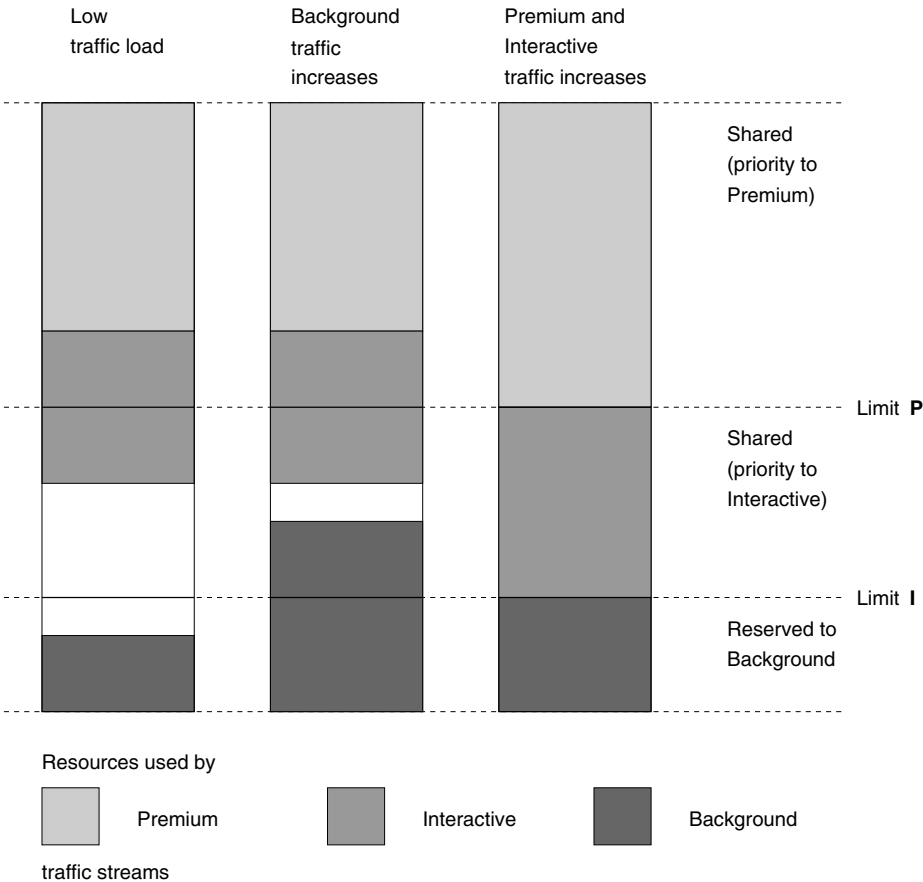


Fig. 4. Admission control policy (example)

3.2 Scheduling in the BSS

Depending on the QoS profile negotiated the BS RLC/MAC layer performs scheduling of the radio blocks. The scheduling mechanism implemented for both uplink and downlink direction follows a three-stage principle (see Figure 5). First, incoming radio blocks are distributed into one of three queues according to the QoS subscription associated with the respective traffic flow. It is differentiated between Premium (“Gold Card”) service, Standard service and Best-effort service. The second stage is only valid for Standard service traffic. Owing to a packet’s application QoS profile, the appropriate traffic class queue is chosen from Conversational, Streaming, Interactive, or Background. Best-effort traffic from the first stage is put into a fifth queue. Within the traffic class queues packets are scheduled according to their TBF and a Round Robin (RR) algorithm with the depth of 20 radio blocks per scheduled TBF in the RR cycle. The third

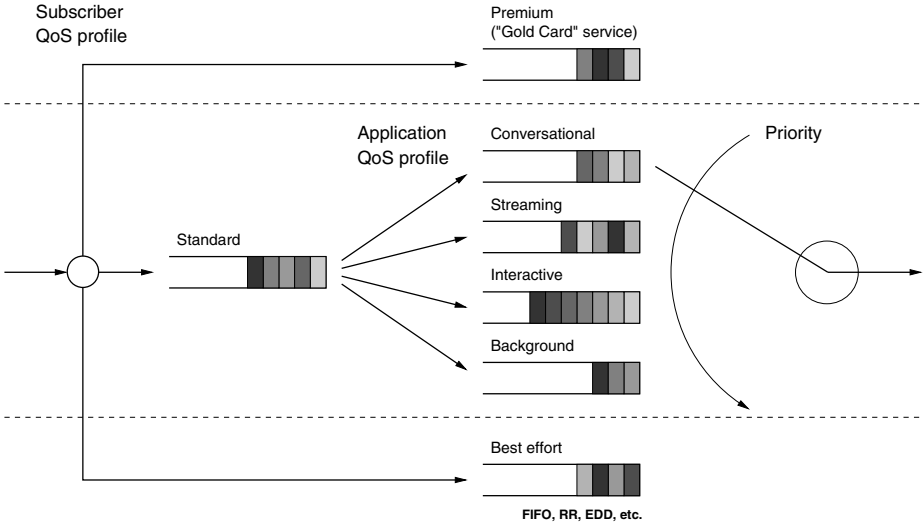


Fig. 5. Principle of the scheduling function located in the BS RLC/MAC layer

stage is built by a simple priority mechanism, serving the traffic class queues in order from highest priority (Premium) to lowest priority (Best-effort).

4 Performance Evaluation

4.1 Simulation Scenarios

The cell configuration is given by the number of transceiver units (TRX) in the radio cell. Here a typical 3-TRX scenario is regarded with 0 and 1 fixed and 8 and 7 on-demand Packet Data Channels (PDCH) that are shared with circuit switched GSM traffic, which is offered corresponding to an Erlang-blocking probability of 1 %. This means that on average around 7 PDCHs are available for GPRS [5].

The channel conditions are determined by a constant RLC/MAC block error probability of 13.5 % corresponding to a C/I of 12 dB. As the coding scheme CS-2 is used.

LLC and RLC/MAC are operating in acknowledged mode. The multislot capability is 4 uplink and 1 downlink slots. The MAC protocol instances in the simulation model are operating with three random access subchannels per 52-frame. LLC has a window size of 16 frames. TCP/IP header compression in SNDCP is performed. TCP is operating with a maximum congestion window size of 8 Kbyte and a TCP Maximum Segment Size (MSS) of 536 byte. The transmission delay in the core network and external networks, i.e., the public Internet, is neglected. This corresponds to a scenario where the server is located in the operator's domain. The session interarrival time is set to 12 seconds. The Internet traffic [6] is composed of 70 % E-Mail sessions and 30 % WWW sessions

(see Table 2) not depending on the subscription profile of the regarded MS. 10 % of the mobile stations are representing Premium subscribers and 90 % Standard subscribers.

Table 2. Model parameters of Internet applications (HTTP and SMTP)

HTTP Parameter	Distribution	Mean
Pages per session	geometric	5.0
Intervals between pages [s]	negative exponential	12.0
Objects per page	geometric	2.5
Object size [byte]	log ₂ -Erlang-k	3700
Amount of SMTP data	Distribution	Mean
E-Mail Size [byte]	log ₂ -normal	10000
Base quota [byte]	constant	300

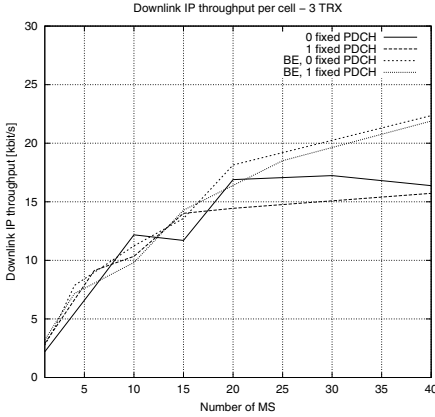
4.2 Performance and System Measures

As performance measures the downlink IP throughput per user during a data transmission and the 95-percentile of the downlink IP packet delay are regarded. These are the QoS measures that are noticed by the user and that can be compared to the ETSI/3GPP QoS classes [3,?]. For WWW and e-mail applications the throughput per user is the important measure since it mirrors the response time of a requested file.

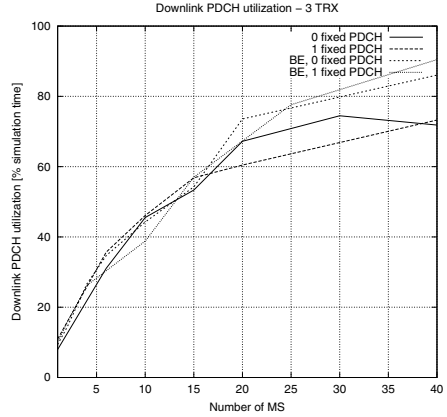
The system measures comprise the downlink IP system throughput per radio cell and the downlink PDCH utilization, which is calculated by the total number of radio blocks carrying data or control information divided by the total number of transmitted radio blocks. The measures are presented over the number of mobile stations (MS) offering GPRS traffic.

4.3 Simulation Results

Figure 6(a) shows the mean downlink IP system throughput per radio cell for 0 and 1 fixed PDCHs and with and without QoS management functions. The difference between the curves with 0 and 1 fixed PDCHs is very small since only in 1 % of the time all PDCHs are allocated for circuit-switched calls. Since the offered circuit-switched traffic is lower for the 1-fixed-PDCH scenario, the system throughput is 1–4 % higher in the 0-fixed-PDCH scenario. As expected the system throughput for low load situations with less than 20 MS in the cell are nearly the same for the results for a best effort (BE) service and a service with QoS functions. In higher load situations the system throughput comes into saturation. This can be explained by the effect that 5 % of the Background sessions are terminated, when no IP packets are received for a period of more

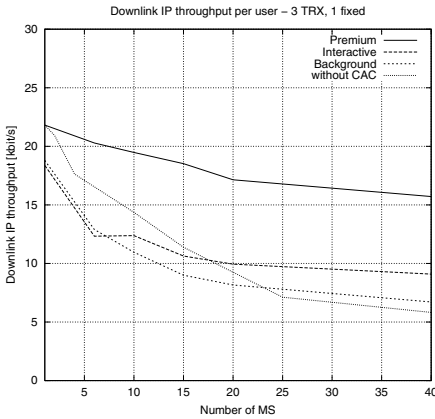


(a) Mean downlink IP system throughput per cell

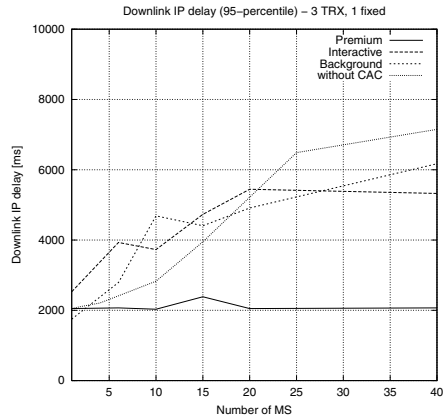


(b) Mean downlink PDCH utilization

Fig. 6. System measures with and without QoS management functions



(a) Mean downlink IP throughput per user



(b) 95-percentile of downlink IP packet delay

Fig. 7. Performance measures for different subscriber and application classes

than 30 seconds. This does not occur in the BE simulations. The same effect can be seen in Figure 6(b), where the channel is not utilized with more than 75 % in the results with QoS functions.

In Figure 7(a) the downlink IP throughput per user during transmission periods for the different service and subscriber classes Premium, Interactive and Background compared with simulation results for a pure BE service (without CAC) is presented. In situations with low traffic load Standard users are losing 15-20 % of performance compared to the BE service while the Premium user performance always remains higher than 15 kbit/s. In higher load situations the service differentiation between Interactive and Background becomes visible. While the throughput performance of the Interactive traffic does not fall under 10 kbit/s, the performance loss for Background applications is not visible in this measure. Nevertheless, 5 % of the Background sessions are terminated because of poor performance as mentioned above. This can be avoided using fairer scheduling algorithms. The 95-percentile of the IP packet delay in Figure 7(b) shows the similar effect.

5 Conclusions

In this paper the capacity and performance gain achieved by quality of service functions in the GPRS radio network comprising Connection Admission Control (CAC) and scheduling with subscriber and service differentiation is examined. Simulation results show that Premium users can be served with nearly constant throughput and delay performance even if the number of active mobile stations in the radio cell rises to 40. 40 Interactive applications instead of 12 in the pure best effort case can be served with a throughput performance of 10 kbit/s, while the performance for Background users remains acceptable even in high load situations. These results show that QoS functions in GPRS networks are increasing the application-specific performance significantly and realize the capability to serve subscribers and applications with respect to their QoS requirements.

References

1. Brasche, G., Walke, B.: Concepts, Services and Protocols of the new GSM Phase 2+ General Packet Radio Service. IEEE Communications Magazine, Volume 35, 1997
2. Walke, B.: Mobile Radio Networks – Networking and Protocols. John Wiley and Sons, Chichester, 2nd Edition, to be published in 2001
3. ETSI 3GPP: Universal Mobile Telecommunications System (UMTS), Service Aspects, Services and Service Capabilities (3G TS 22.105, Version 3.8.0, Release 1999)
4. ETSI TC-SMG: Digital Cellular Telecommunications System (Phase 2+), General Packet Radio Service (GPRS), Service description, Stage 2 (GSM 03.60, Version 7.4.0, Release 1998)
5. Stuckmann, P., Müller, F.: GPRS Radio Network Capacity and Quality of Service using Fixed and On-Demand Channel Allocation Techniques. Proc. of the Vehicular Technology Conference Spring 2000
6. Stuckmann, P., Seidenberg, P.: Quality of Service of Internet Applications over GPRS. Proc. of the European Wireless'99, ISBN 3-8007-2490-1

Case Studies and Results on the Introduction of GPRS in Legacy Cellular Infrastructures

C. Konstantinopoulou, K. Koutsopoulos, P. Demestichas, E. Matsikoudis
and M. Theologou

National Technical University of Athens (NTUA), Dept. of Electrical and Computer Eng.,
9, Heroon Polytechniou Str., 157 73, Zographou, Athens, Greece
ckonsta@telecom.ntua.gr

Abstract. The aim of this paper is: (a) to define suitable traffic source models for representing various types of data services/applications and (b) to study the performance resulting from the introduction of GPRS services in legacy infrastructures, through a generic simulation platform which takes into account air-interface, access and backbone network configuration aspects. The performance will be studied for a range of voice traffic loads, number of TRXs, offered services, number of dedicated PDCHs, etc.

1 Introduction

Data services like Web browsing, e-mail and file transfer are becoming more and more popular in cellular systems. Up to now, GSM¹ data transfer has been Circuit-Switched (CS) i.e., physical resources are allocated to a user for the entire service session. However, this is insufficient in case of bursty traffic, where bursts are separated by long intervals of inactivity. This has been the main reason for the introduction of General Packet Radio Service (GPRS), which on the one hand acts as a mobile access network to the Internet, while on the other hand it enables the operator to offer a wide variety of value-added services efficiently (WAP over GPRS, e-banking, e-commerce, push services, etc.) [1]. The aim of this paper is two-fold: (a) to define suitable traffic source models for representing various types of data services/applications and (b) to study the performance resulting from the introduction of GPRS services in legacy infrastructures, through a generic simulation platform which takes into account air-interface, access and backbone network configuration aspects. The performance will be studied for a range of voice traffic loads, number of TRXs, offered services, number of dedicated Packet Data Channels (PDCHs), etc.

¹ Global System for Mobile communications

2 Simulation Platform

A generic GPRS simulation platform composed of four components (Fig. 1) has been developed [2]:

1. **GPRS Environment Representation (GER):** Comprises libraries with traffic source models representing the service session behavior in terms of session duration, packet calls/session in uplink/downlink, QoS (i.e., throughput, bitrate, session dropping, call blocking) in conjunction with user behavioral characteristics.
2. **GPRS Network Representation (GNR):** It comprises models that correspond to GPRS network elements (i.e. SGSN, GGSN, BG, CG), while the GSM traditional elements (BTSs, BSCs, MSC/VLRs, HLR, etc.) incorporate GPRS related functionality (i.e., dynamic allocation of radio resources to CS and PS traffic).
3. **GPRS Simulator Event Scheduler (GSES):** It provides the means for representing, storing, and manipulating (inserting, retrieving, deleting, etc.) the events that the core of the simulator will have to handle.
4. **GPRS Simulator Controller (GSC):** It is responsible for the initialization of GNR, GER and GSES and handles the communication between them.

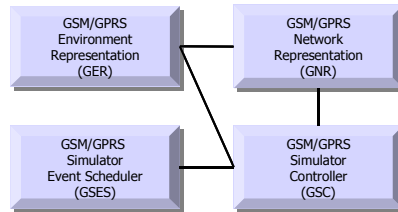


Fig. 1. GSM/GPRS Simulation Platform

3 Traffic Source Models

The definition of a generic source model “representing” the service classes as defined in [3] (conversational, streaming, interactive² and background³), is very difficult. Since our studies focus on GPRS, only interactive and background service classes have been considered.

3.1 Interactive Services

A data session [4] consists of a sequence of packet calls, while a packet call is composed of a bursty sequence of datagrams (Fig. 2).

² Interactive services are typical instant response request services (i.e., web browsing, data base retrieval, server access, polling for measurement records and automatic data base enquiries).

³ In background services the end-user sends and receives data in the background (i.e., background delivery of e-mails, download of databases).

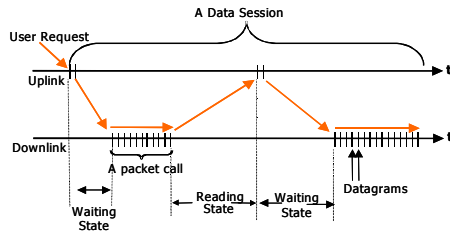


Fig. 2. Typical Characteristics of an Example Interactive Service

A detailed source model for interactive services is depicted in Fig. 3. As shown, four states can be identified:

1. Active State (user active, host inactive): The MS sends a “packet uplink channel request”. If a collision has occurred, the MS retries using backoff. On no collision, the MS is assigned the uplink PDCHs that will be used for uplink transmission.
2. Waiting State (user inactive, host inactive): After the successful transmission of the uplink request, the MS waits for the assignment of the downlink PDCHs, through which the requested information will be transmitted.
3. Receiving Information State (user inactive, host active): A “packet downlink assignment message” is sent to the MS comprising the assigned downlink PDCHs.
4. Reading State (user inactive, host inactive): Represents the time needed for processing the downloaded information before making the new request.
- 5.

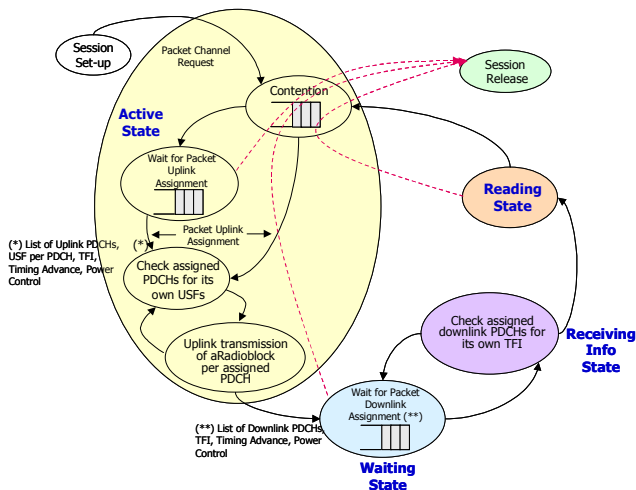


Fig. 3. Traffic Source Model for Interactive Services

3.2 Background Services

The uplink transmission for background upload services is similar to the interactive services one. The difference is that the MS performs the “packet uplink channel request” only once and the uplink transmission is performed through the assigned

PDCHs. In case of download background services, after the successful uplink request transmission, the MS receives a “packet downlink assignment” message and starts monitoring the assigned downlink PDCHs to identify the packets addressed to it.

3.3 Source Models Parameters

Table 1 illustrates the traffic source models parameters indicating in which type of service these are applicable.

Table 1. Traffic Source Models Parameters

Services Characteristics	Description	Interactive	Download Back ground	Upload Back ground
#packet calls/session (N_{pc})	Exponentially distributed with a mean μ_{Npc}	✓	✓	✓
#datagrams within a packet call (N_d)	Follows Pareto distribution with cut-off	✓	✓	✓
Datagrams interarrival time (D_d)	Exponentially distributed with a mean μ_{Dd}	✓	✓	✓
Datagram size, (S_d)	Uniformly distributed variable	✓	✓	✓
GPRS backbone and Internet delay (D_{GI})	Exponentially distributed with a mean μ_{DGI} . Represents the time interval between a successful transmission of an uplink request and the arrival of the first datagram of the downlink packet call	✓	✓	N/A
Reading time (D_{pc})	Exponentially distributed with a mean μ_{Dpc}	✓	N/A	N/A
User Behavior				
User tolerance time (T_{ut})	Constant variable; models user behavior.	✓	N/A	N/A
Max. #user retries (N_{ur})	Max. #user retries when user tolerance time has expired	✓	N/A	N/A

4 Case Studies

1. **CASE STUDY 1:** Estimation of the maximum CAR for each offered data service for a range of voice traffic loads, while maintaining the QoS of the CS and PS connections under tolerable levels.
2. **CASE STUDY 2:** Estimation of the required number of TRXs in a cell, given specific operator's requirements (CARs, services characteristics, etc.).
3. **CASE STUDY 3:** Investigation of the impact of dedicated PDCHs on the overall system performance.

5 Results

The performance of a GPRS-capable network under various loading conditions and for various data services is investigated. The system performance is characterized by the max. CAR/data service that the system can support (for specific data services characteristics, voice traffic and TRX availability) provided that the QoS criteria are respected. Note that: (a) for interactive services abnormal session release (session dropping) occurs whenever the maximum number of user retries has been reached, while (b) for background services, abnormal session release may be initiated (either by the system or the user) whenever the “negotiated QoS” falls below certain limits e.g., the offered mean bitrate falls below a certain percentage (70%) and remains there for a certain time period (20 sec).

5.1 Case Study 1

5.1.1 Input Data

#cells	: 1
#TRXs/cell	: 4
#Traffic Channels (CS and PS)	: 30
Mean call duration	: 55 sec
Services characteristics	: Table 2
Voice call blocking probability	: 1%
Session blocking probability	: 1%
Session dropping probability	: 2%
Voice CAR	: 60-1320 calls/h
Coding Schemes used	: CS1-CS2

Table 2. Services Characteristics

SERVICES CHARACTERISTI CS	Web- browsing Uplink	Web- browsing Downlink	Download e- mail	Upload e- mail
N _{pc}	30	30	8	8
N _d	5	40	20	20
S _d (bytes)	200-500	200-700	300-700	300-700
D _{pc} (sec)	N/A	40	N/A	N/A
D _{GT} (msec)	40	N/A	40	N/A
D _d (sec)	0.001	0.001	0.001	0.001
Max & Min PDCH up- and downlink	1	Min:1-Max:3	Min:1-Max:2	Min:1-Max:2
USER BEHAVIOR				
T _{ut} (sec)	120	N/A	N/A	N/A
N _{ur}	3	N/A	N/A	N/A

5.1.2 Results

The results concern:

1. The maximum CAR of the data services that the system can withstand for a range of voice CARs. The results indicate that the CAR for data sessions decreases, as voice load increases. We also observe that e-mail CAR (upload and download) is higher than the Web-browsing one. This is justified by the fact that the e-mail is less demanding than Web-browsing since: (a) it requires less slots and (b) each session consists of a low number of packet calls, leading thus to shorter sessions durations. Moreover, the upload e-mail CAR is higher than that of download e-mail, as the downlink is dominated by Web-browsing downlink traffic and download e-mail.

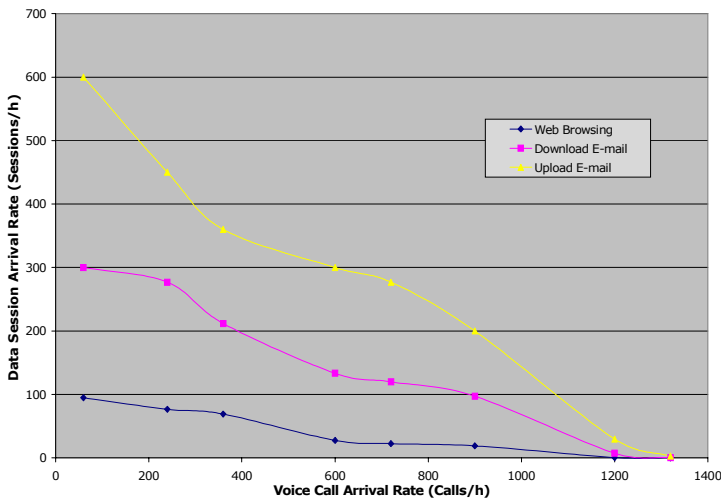


Fig. 4. Max. Supported CAR for Offered Data Services versus Voice CAR

2. The average number of slots allocated to data services and the mean bitrate for various voice CARs. According to the obtained results, the requested PDCHs for both uplink and downlink are actually assigned for medium and high voice traffic loads, while for low voice traffic (where the number of data sessions is high) the above is not always satisfied (downlink Web-browsing, download e-mail). Obviously the latter affects the services mean bitrate.
3. The percentage of slot utilization in uplink and downlink for both GSM and GPRS services show that the maximum achievable slot utilization for voice only service (GSM) is 60.9% for both uplink and downlink, while for a combined GSM/GPRS network, the maximum achievable slot utilization in uplink and downlink is 63.95% and 77.78% respectively. The difference in uplink and downlink slot utilization is justified by the asymmetric nature of the assumed services.

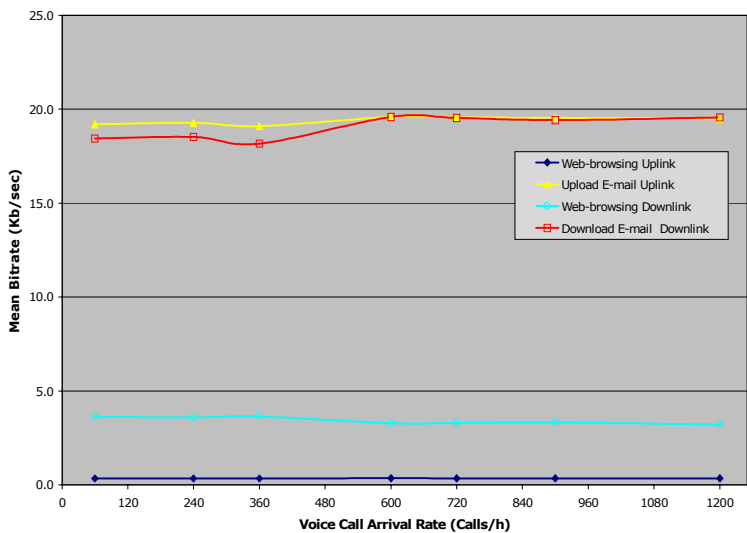


Fig. 5. Mean Bitrate (kbits/sec) for Various Voice CAR (Calls/h)

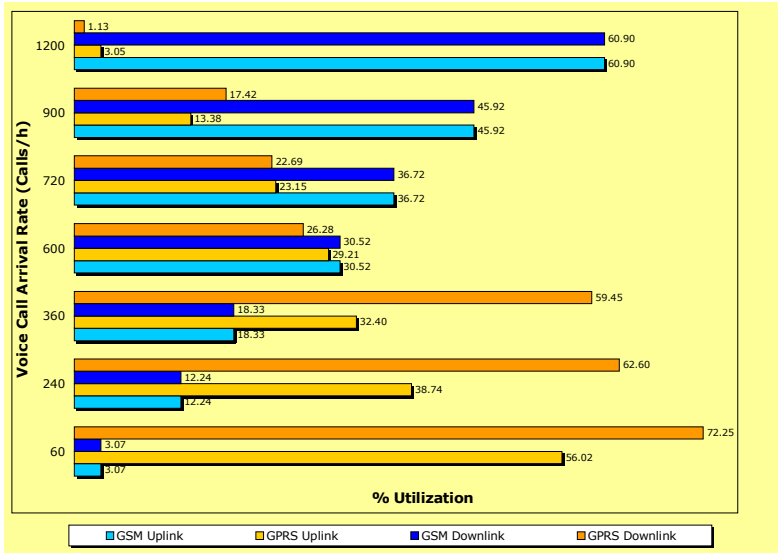


Fig. 6. Slot Utilization in Up-Downlink for GSM/GPRS Services for Various Voice CARs

5.2 Case Study 2

5.2.1 Input Data

The same as in case study 1.

Operator's specific data:

Voice traffic load	: 13.75 Erlangs
Web-browsing CAR	: 0.45 calls/sub/h
Download e-mail CAR	: 1.5 calls/sub/h
Upload e-mail CAR	: 2.5 calls/sub/h
%GSM subscribers that are GPRS subscribers too	: 15%
%GPRS attached subscribers during busy hour	: 75%
%GPRS attached subscribers having an active PDP context during busy hour	: 95%

5.2.2 Results

The results show that 4 TRXs are not enough for satisfying operator's requirements (especially for Web-browsing), while 5 TRXs are more than enough as in this case the maximum supported Web-browsing CAR is 0.773 calls/sub/h.

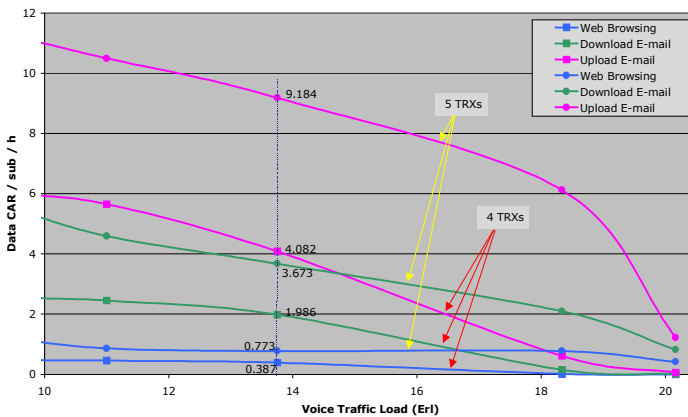


Fig. 7. Max. Supported CAR for Offered Data Services/Subscriber/Hour versus Voice Traffic

5.3 Case Study 3

5.3.1 Input Data

- #Dedicated PDCHs: 0–4
- Voice CAR=1200 calls/h
- The rest is as in case study 1.

5.3.2 Results

We observe that as the number of dedicated PDCHs increases, both the CAR of data sessions and the voice call blocking probability increase. The data CAR increase is justified by the fact that at high voice traffic load, packet calls can still be supported efficiently due to the existence of dedicated PDCHs. The increase of the voice call blocking probability above 1% when assuming 3 and 4 dedicated PDCHs, is justified by the fact that the traffic channels that could be utilized by CS (voice) connections ($30-3=27$ and $30-4=26$ respectively) cannot satisfy the criterion of 1% blocking (according to Erlang-B formula) for the specific voice CAR (1200 calls/h).

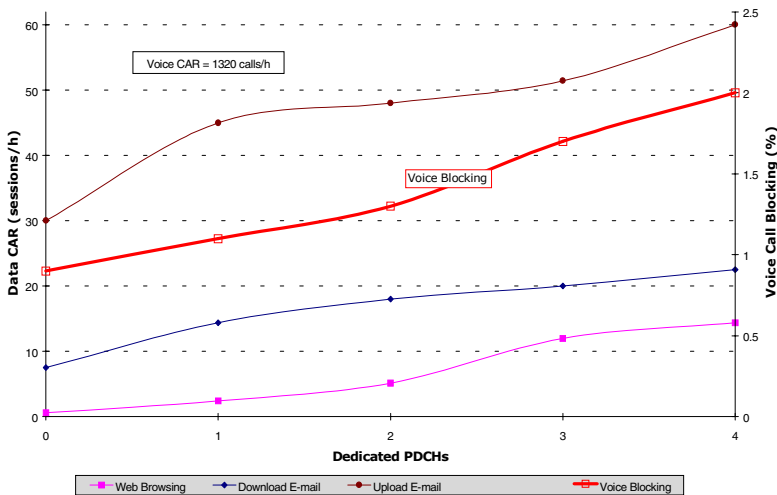


Fig. 8. Max. Supported CAR for Offered Data Services versus #Dedicated PDCHs

6 Conclusions

In this paper we have investigated the performance of a combined GSM/GPRS capable network for a range of voice traffic loads, number of TRXs, offered services (both interactive & background ones), number of dedicated PDCHs, etc. The results have shown that: (a) Legacy cellular systems can withstand a significant volume of PS data traffic, (b) The introduction of GPRS leads to better slot (radio resource) utilization compared to a pure GSM network and (c) The use of dedicated PDCHs improves the quality levels provided to data sessions, but at the same time, at high voice traffic loads the call blocking probability may increase above tolerable levels (1%).

References

1. J. Cai, D.J. Goodman, "General packet radio service in GSM", *IEEE Commun. Mag.*, Vol. 35, No. 10, Oct. 1997
2. K. Koutsopoulos, C. Konstantinopoulou, D. Bidinost, S. Kotrotsos, N. Alexiou, P. Demestichas and M. Theologou, "Issues in the Provision of Wireless Access to Internet Services through GPRS", in Proc. *IST Mobile Communications Summit 2000*, Galway, Ireland, Oct. 2000
3. "QoS Concept and Architecture", *3GPP Technical Specification Group Services and System Aspects; 3GPP TS 23.107, V3.4.0 (2000-10) (Release 1999)*
4. "Universal Mobile Telecommunications System (UMTS) Selection procedures for the choice of radio transmission technologies of the UMTS", *ETSI UMTS 30.03 version 3.2.0*.

Traffic Analysis of Multimedia Services in Broadband Cellular Networks

Péter Fazekas¹ and Sándor Imre¹

Budapest University of Technology and Economics,
Department of Telecommunications,
Pázmány Péter s. 1/D, H-1117 Budapest, Hungary,
{fazekasp,imre}@hit.bme.hu

Abstract. The enormous development of wireless multimedia communication techniques requires new modeling methods. In this paper a novel analytical technique is presented, to examine wireless networks with multiple connection types that may change their generated traffic in time. By our method call-level system parameters can be calculated. We also propose two simple base station admission control policies and investigate the effect of these policies and the effect of reserved capacity for several connection types.

1 Introduction

One of the major directions of today's telecommunications research is the integration of high speed data communications with wireless access. Considering the rapidly increasing computational capacity of portable computers, the possibility of using broadband multimedia applications is already given in mobile nodes. However, wireless networks are currently not capable of providing high data rates for such applications.

The evolution of second generation mobile systems towards the third generation (UMTS, IMT2000 family) assures that packet switched data and multimedia communications are beginning to spread instead of voice-oriented mobile systems. Yet the transmission speed of wired LANs will not be available in 3G systems.

Besides the development of 3G systems enormous research is being carried in the field of wireless LANs and other broadband cellular networks. These systems are not intended to provide wide range coverage, rather the aim of these is similar to that of wired LANs. These radio networks (e.g. Hiperlan 2, broadband extension of 802.11, wireless ATM LANs) are capable of providing several tens of Mbps aggregated transmission speed. In such wireless networks any application can run. These applications often do not generate traffic at constant speed, thus treating them as constant bit-rate sources is not appropriate.

As we have seen both the industry of commercial cellular systems and wireless LANs move towards the direction of carrying heterogeneous, multimedia traffic with reasonable bandwidth. For preliminary network design and network

dimensioning analytical models are needed that describe the behavior of the system considering connections that generate variable amount of traffic in time. To use the radio spectrum efficiently wireless MAC protocols are needed that allocate to a user radio resources according to its instantaneous need.

Recently a number of papers proposing models of cellular networks with multiple service classes were published (e.g. [1]-[3] and references therein). Despite the anticipatory significance of wireless multimedia communications these analytical models of wireless networks do not take into account the versatility of a customer's occupied capacity. Even in the case when multimedia connections are considered, their required capacity is characterized by the - constant - effective bandwidth.

In most of the literature a customer is characterized by three time variables. One of these is the dwell time which is the time a mobile spends within a radio cell while it is roaming throughout the area. The session length is the time a connection lasts regardless the terminal's mobility. This means that the above two variables are independent. The third variable is the channel holding time, that a connection spends actively in a cell. This is usually calculated from the first two.

Both the session length and the dwell time is often modeled with exponential distribution, although this assumption is only valid under specific circumstances. Recently more sophisticated dwell time and session length distributions were suggested (e.g. [2]-[4]).

In this paper we propose a modeling framework for calculating performance parameters of cellular networks when customers are present that change their amount of occupied capacity during a connection. The model is based on a Markovian description of a single base station considered as a pool of a given amount of shared capacity. We present an approximate method that is appropriate for the calculation of new call blocking probability, handover failure probability and channel utilization.

This paper is organised as follows. In Section 2 and 3 the general description of the modelling environment and the derivation of the service time distribution is given. In Section 4 and 5 the abstract model of the base station and its solution method is investigated. This is followed by numerical results and conclusions.

2 Mobile User Model

The method presented here focuses on a single base station as if it were functioning in isolation. Sessions can admit to the base station as new calls initiated within the coverage area of the base station (referred as new connection), or as a handover attempt from the vicinity of the cell (handover call). We suppose that the arrival of new calls and handover calls are appropriately described as two Poisson processes.

We assume that there are K customer types present in the system. Customers belonging to different types may belong to different service classes, in this case the characteristics of their generated traffic is different as well as probably their

session lengths. Customers that belong to the same service class may belong to different types in the case when their mobility behavior is different, namely when their dwell times are different. As we show in the next subsection we must distinguish between handover calls and new calls as well. In this paper the super-, or subscript H means handover call, while N denotes new call.

2.1 Residual Dwell Time and Session Length

Let us denote the dwell time and the session duration of a type k customer by τ_D^k and τ_S^k respectively. When examining a single base station, the knowledge of the dwell time and the session duration of each user is not sufficient. Rather we introduce the notion of residual dwell time and the residual session length to describe the difference between handover calls and new connections, both are derivatives of the dwell time and the session duration. The residual session length of handover connection of class k is denoted by $\tau_{R,S,H}^k$; this is the period between the call's admission to the base station and the termination of the call, regardless the place of call termination. For connections initiated within the cell it obviously follows from the definition that the residual session length is equal to the session length, i.e. $\tau_{R,S,N}^k = \tau_S^k$. This also holds if the session length is exponentially distributed, due to the memoryless property of the exponential distribution. The residual dwell time of a type k connection that is initiated in the cell is denoted by $\tau_{R,D,N}^k$. This is the time interval between the time instant the customer starts transmission in the cell and leaves the cell, regardless it has terminated its call or not. It is again obvious from the definition, that for handover calls this residual dwell time equals to the dwell time and this equation holds for new calls as well if the dwell time is exponentially distributed.

The derivation of the residual session length and dwell time is completely out of the scope of this paper, therefore we suppose that the distributions of the four descriptor times of each customer types or at least statistics of these times are given. Our analysis requires each descriptor distribution to be approximated by Phase Type distributions (PH). A phase type distributed time is a mixture of a number of exponentially distributed phases [6]. In other terms, the PH is the time a finite state Markov chain reaches an absorbing state. The PH is characterized by the initial probability of the Markov chain, the rates among states and the rates between each state and the absorbing state.

Studies show (e.g. [5] and references therein) that with an appropriately chosen PH any distribution can be approximated with arbitrary accuracy. Moreover, if a series of statistical data of a random variable is available, its (unknown) distribution can be accurately approximated with a proper PH.

To approximate user describing times with PH has the advantage that while it enables arbitrary distributions to characterize customer behavior it still exploits the memoryless nature of the exponential distribution (the phases of the PH last for exponentially distributed time), thus standard queuing analysis techniques can be used to examine the system at the expense of larger state space.

The versatility of a customer's generated data is fulfilled by means of a finite Markov chain. Each state of the chain is assigned with a given transmission

rate. This means that the active user transmits with a certain bit-rate for an exponentially distributed time, than it changes its transmission speed according to the describing Markov chain. This is a widely accepted model of VBR traffic [7]-[8] and with appropriately chosen Markov chain the generated traffic of a VBR connection can be characterized correctly. The probability that a connection starts transmission with a given rate is described by the initial probability vector of the Markov chain. In this paper the term variable bit-rate and the abbreviation VBR does not mean the VBR class of ATM, but simply a connection type that changes its transmission speed during a connection. Naturally constant bit-rate connection types may also be present in the system, the model is applicable for such types without any modification.

Because a handover connection is already active when arriving, its initial probability vector of the VBR describing Markov chain is not the same as for a new call. It is reasonable to suppose that before handover the session was active for enough time so the Markov chain reaches the equilibrium. Thus the initial probability of the VBR Markov chain of handover calls is supposed to be equal with the steady-state distribution of the chain.

2.2 Base Station Model and Call Admission

The base station is modeled as a channel pool of C_0 units of capacity. In general the blocking of a handover attempt is less tolerable then blocking a new call initiation. Furthermore we suppose that there are some connection types that tolerate call blocking less than others. Therefore the base station shares its capacity among connection types not equally: the available capacity of a type k handover call is denoted by C_k^H , that of type k new call is C_k^N .

In this paper we investigate two simple base station admission control policies that can be applied if there is not enough capacity to admit the connection with its actual transmission speed:

- policy 1: the call with too high instantaneous transmission speed is immediately blocked,
- policy 2: the connection is not blocked, but the base station forces it to begin its transmission with a lower rate. The call begins transmitting with the highest among its possible rates that can be admitted to the base station. The call is only refused when the available capacity at the base station is less than the lowest possible transmission rate of the connection. Although applying this policy the connection attempt is not immediately refused, but since it has to transmit with a lower rate that it intended, the call may suffer some degradation of its QoS parameters. For each connection type and handover and new connection any of the two policies may be applied.

Due to the existence of VBR connections, the amount of occupied capacity may change without the initiation of new or handover call or call termination. In the case when a mobile tries to switch to a higher bit-rate but there is not enough free capacity to do this, it is reasonable to assume that the call is not dropped but the user is able to continue its transmission with the previous rate.

3 Service Time Distribution

The aim is to derive a queueing model of the presented system. The input process is assumed to be Poisson and a service time distribution is needed to create a Markov chain description of a single base station.

The service time is the time a connection spends in the system communicating, i.e. its distribution is equal to the distribution of the channel holding time. From the definitions of the residual dwell time and session length, it is clear that the channel holding time is the following:

$$\tau_{C,H}^k = \min(\tau_{R,S,H}^k, \tau_D^k), \quad \tau_{C,N}^k = \min(\tau_{R,D,N}^k, \tau_S^k) \quad (1)$$

where $\tau_{C,H}^k$ is the channel holding time for a type k customer that arrived with handover and $\tau_{C,N}^k$ is the channel holding time for type k calls initiated in the cell.

Let the descriptors of the PH distributed dwell time of type k users be denoted by $\mathbf{D}^k$, $\underline{d}^k$ and $\underline{D}^{0,k}$, where $\mathbf{D}^k = \{D_{ij}^k\}$ contains the rate from phase i to phase j in its i, j -th position (nor i , nor j is the absorbing state), the i th element of $\underline{d}^k$ is the probability that the PH random time begins with phase i and the vector $\underline{D}^{0,k}$ contains the rates from each phase to the absorbing state (finish) of the process. According to the properties of Markov chains, $D_i^{0,k} = -\sum_j D_{ij}^k$. Similarly, the parameters of τ_S^k are $\mathbf{L}^k$, $\underline{l}^k$ and $\underline{L}^{0,k}$, that of $\tau_{R,S,H}$ are $\mathbf{L}^{k,H}$, $\underline{l}^{k,H}$ and $\underline{L}^{0,k,H}$ and $\tau_{R,D,N}$ is characterized by $\mathbf{D}^{k,N}$, $\underline{d}^{k,N}$ and $\underline{D}^{0,k,N}$.

The channel holding time of a new call can be composed from the PH distributed $\tau_{R,D,N}^k$ and τ_S^k as follows. The phases of the session duration time is taken as many times as many phases the residual dwell time has, with the rates of $\mathbf{L}^k$ among the phases within a group. Between the appropriate states of these groups the rate is equal to the rates between the corresponding phases of the residual dwell time.

To track the occupied capacity at the base station accurately the service time must contain information on the actual bandwidth requirement of the connection. To achieve this the service time is composed from the channel holding time and the VBR describing Markov chain the same way as the channel holding time was composed. During this procedure the role of dwell time is replaced with the VBR describing Markov chain and the role of call length with the channel occupancy distribution. The rate matrix of the VBR Markov chain is denoted by $\mathbf{V}^k$, the number of possible transmission rates is V^k and its initial vector is $\underline{v}^{k,H}$ for handover calls and $\underline{v}^{k,N}$ for connections initiated in the cell.

The service time composed the described way also has a PH distribution, with the descriptors:

$$\begin{aligned} \mathbf{S}^{k,N} &= \mathbf{Q}^k \oplus (\mathbf{D}^{k,N} \oplus \mathbf{L}^k), \quad \underline{S}^{0,k,N} = h_{V^k} \otimes (\underline{D}^{0,k,N} \oplus \underline{L}^{0,k}), \\ \underline{s}^{k,N} &= \underline{v}^{k,N} \otimes (\underline{d}^{k,N} \otimes \underline{l}^k), \end{aligned} \quad (2)$$

where $\oplus$ and $\otimes$ denotes the Kronecker sum and product, respectively.

It is easy to show that the distribution of the PH composed in the described manner is the same as the distribution of $\min(\tau_{R,D,N}^k, \tau_S^k)$, i.e. this is indeed

the channel holding time distribution and it also contains the instantaneous transmission rate of the connection.

The service time for handover calls can be composed analogously. Since we assume that an call does not change its type during transmission, the $2K$ service times generated (one for handover and newly generated calls of all K types) can be handled independently.

4 Blocking Probabilities

We showed that the service time of this system with C_0 servers has a PH distribution given by (2). Since the input process is Poisson formally we have an M/PH/ C_0 queue with phase dependent capacity requirements.

The state of the describing process is the vector $\underline{n} = (\underline{n}^{1,N}, \dots, \underline{n}^{K,N}, \underline{n}^{1,H}, \dots, \underline{n}^{K,H})$, where $n_i^{k,H}$ denotes the number of type k users arrived with handover receiving phase i of their service. Let $\underline{c}^{k,N}$ denote the vector containing the capacity demands associated with the phases of the service time of a type k connection that is initiated in the cell. The valid states of the system are those, where for any k :

$$\underline{n}^{k,N} \cdot \underline{c}^{k,N} \leq C_k^N, \quad \underline{n}^{k,H} \cdot \underline{c}^{k,H} \leq C_k^H, \quad \sum_{k=1}^K \underline{n}^{k,N} \cdot \underline{c}^{k,N} + \underline{n}^{k,H} \cdot \underline{c}^{k,H} \leq C_0. \quad (3)$$

This simply means that the total amount of occupied capacity can not be larger than the maximum capacity and each connection type can occupy less then its available capacity.

In [11] we described the state space and the driving process of the proposed system. We also pointed out that due to the multiple dimensions of the state space the resulting Markov chain may have several millions of states, therefore calculating its steady state distribution as the solution of the $\underline{pQ} = 0$ global balance equations is impossible.

To calculate blocking probabilities and channel utilization the steady state distribution of the system is not required. Rather the channel occupancy probabilities are needed, i.e. $p(m) = Pr\{\text{m units of capacity is occupied in the base station}\}$.

Given the channel occupancy distribution, the performance parameters of the system can be calculated as follows. The call blocking probability in case of applying policy one for a type k call initiated in the cell is:

$$p_B^N = \alpha_k \cdot \sum_{i=1}^{V^k} \sum_{m=C_k^N - c_i^{k,N}}^{C_0} p(m), \quad (4)$$

where V^k denotes the number of possible transmission speeds of a type k customer and α_k is the probability that an arriving connection is of type k . The same measure for handover calls is calculated analogously.

If we denote the minimum possible capacity requirement of a type k call with c_{min}^k , the call blocking probability for a type k call initiated in the cell applying the second service policy has the form of:

$$\hat{p}_B^N = \alpha_k \cdot \sum_{m=C_k^N - c_{min}^{k,N}}^{C_0} p(m). \quad (5)$$

The channel utilization is simply given as:

$$\varrho = \sum_{m=0}^{C_0} m \cdot p(m). \quad (6)$$

5 Approximate Channel Occupancy Calculation

If we assume the base station capacity C_0 to be infinitely large, the Markov chain of the problem has the nice property that its equilibrium distribution has a product form. This means that each state's probability can be calculated as the product of one of its neighbor's probability and a given factor. Moreover, if all the capacity demands were the same (CBR connections), this property would hold even in case of finite base station capacity.

The Markov chain has a product form solution if and only if local balance equations apply throughout the whole state space. We observed, that in our problem at the majority of the state space the local balance equations hold. The local balance equations change in those states that represent an amount of total occupied capacity that is too big, so some connections can not be admitted or the raise of transmission speed is not possible. We refer to these states as the states of a blocking sub-space. In these blocking sub-spaces local balance equations also hold, but have different form comparing to the equations of the non-blocking space. Therefore we conclude that the form of local balance equations depends on the total occupied capacity of the base station, which is a key observation for further analysis. This means that the multiplying factors that appear in the product form solution also depend on the occupied capacity.

Kaufman [9] and Roberts [10] proposed a recursive formula to compute channel occupancy distribution in a shared channel. They considered connections with constant capacity requirements. Their method provides exact values in case of a product form Markov chain. Their method is based on the mapping of the state space into a one dimensional space and they use the multiplying factors in their recursive formula to calculate channel occupancy probabilities.

By examining base stations with infinite capacity we realized, that at the non-blocking sub-space the rates into a phase of the service time due to arrivals and transmission rate changes hold the local balance with the rates out of the phase due to call termination or again transmission rate change. Since no transition is possible between phases of service time distributions of different connection types, it is enough to formulate the local balance equations for a type k connection that was initiated within the cell, for new calls and other

type calls the local balance equations are formulated analogously. The number of customers receiving different phases of the service time is described by $\underline{n}^{k,N}$, but for sake of simplicity in this derivation we denote this with $\underline{n}$, the local balance equations has the form of:

$$\lambda_N \alpha_k s_i^{k,N} p(\underline{n}) + \sum_{j,j \neq i} p(\underline{n} + \underline{e}_j) S_{ji}^{k,N} \cdot (j+1) = p(\underline{n} + \underline{e}_i)(i+1) \cdot (S_i^{0,k,N} + \sum_{j,j \neq i} S_{ij}^{k,N}) \quad (7)$$

Since the service time has a PH distribution $S_i^{0,k,N} + \sum_{j,j \neq i} S_{ij}^{k,N} = -S_{ii}^{k,N}$. Writing the equations into vector form, we get:

$$-\lambda_N \alpha_k \underline{s}^{k,N} p(\underline{n}) = \{(n_1 + 1)p(\underline{n} + \underline{e}_1) \cdots (n_i + 1)p(\underline{n} + \underline{e}_i) \cdots (n_P + 1)p(\underline{n} + \underline{e}_P)\} \mathbf{S}^{k,N}, \quad (8)$$

where for the sake of simplicity, the number of phases of the service time ($D^{k,N} \cdot L^{k,N} \cdot V^k$) is denoted by P . Introducing the vector

$$\underline{F}^{k,N} = \left(\frac{(n_1 + 1)p(\underline{n} + \underline{e}_1)}{p(\underline{n})}, \dots, \frac{(n_P + 1)p(\underline{n} + \underline{e}_P)}{p(\underline{n})} \right),$$

we have

$$\underline{F}^{k,N} = -\lambda_N \cdot \alpha_k \cdot \underline{s}^{k,N} \cdot (\mathbf{S}^{k,N})^{-1}. \quad (9)$$

The vector defined by (9) plays the role of the multiplying factor between the probabilities of two neighboring states.

In this approach a blocking state can be viewed as if the service time was changed. This means that since in a blocking states there are some “unreachable” phases of the service time, thus blocking states result in the change of the service time descriptors, $\underline{s}^{k,N}$ and $\mathbf{T}^{k,N}$. I.e those elements of $\mathbf{T}$ that represent an illegal phase transition because of lack of capacity are set to 0 and the diagonal elements of $\mathbf{S}^{k,N}$ are updated such that $\sum_j S_{ij}^{k,N} + S_i^{0,k,N} = 0$. If admission policy 1 is applied, the arrival of calls with a capacity demand higher than the available bandwidth is restricted, thus those elements of $\underline{s}^{k,N}$ are set to 0 as well.

Thus from this point of view the descriptors of the service time distribution depend on the occupied capacity $x = \underline{n} \cdot \underline{c}$. Thus the change of service times can be formulated as:

$$s_{ij}^{k,N}(x) = 0, \quad s_j^{k,N} = 0 \quad \forall j : c_j^{k,N} > C_k^N - x \quad (10)$$

If policy 2 is applied, the service time changes as follows:

$$S_{ij}^{k,N}(x) = 0, \quad \forall j : c_j^{k,N} > C_k^N - x \quad (11)$$

and if $c_i^{k,N}$ is the maximum capacity demand such that $c_i^{k,N} \leq C_k^N - x$ then

$$s_i^{k,N}(x) = \sum_{j: c_j^{k,N} > C_k^N - x} s_j^{k,N} + s_i^{k,N}(0). \quad (12)$$

Then the factor of (9) depends on the occupied capacity as well, therefore (9) gets the general form of

$$\underline{F}^{k,N}(x) = -\lambda_N \underline{s}^{k,N}(x) (\mathbf{S}^{k,N}(x))^{-1}. \quad (13)$$

Given these equations supposing local balance, the approximate numerical calculation of the blocking probability follows the pattern proposed by Kaufman and Roberts [9],[10]. Define $\tilde{p}(m)$ and $v(p)$, the relative and the normalized probability of that m amount of capacity is occupied in equilibrium. $\tilde{p}(m)$ is computed as $\tilde{p}(m) = 0$ for $m < 0$, $\tilde{p}(0) = 1$, and for $m > 0$

$$\begin{aligned} \tilde{p}(m) = \sum_{k=1}^K \sum_i \tilde{p}(m - c_i^{k,N}) \frac{c_i^{k,N}}{m} F_i^{k,N}(C_0 - m + c_i^{k,N}) + \\ \tilde{p}(m - c_i^{k,H}) \frac{c_i^{k,H}}{m} F_i^{k,H}(m - c_i^{k,H}) \end{aligned} \quad (14)$$

and

$$p(m) = \tilde{p}(m) \frac{1}{\sum_{m=0}^{C_0} \tilde{p}(m)}. \quad (15)$$

Finally, the blocking probabilities and the channel utilization are obtained as (4)–(6).

We intuitively feel, that this method has better accuracy if the system is under light load conditions, meaning that the blocking sub-spaces has negligible probabilities compared to the non-blocking space.

6 Numerical Results

6.1 Accuracy of the Proposed Approximation

As we described previously we expect that our method gives more accurate results under light load conditions and its accuracy deteriorates if the offered traffic increases. To give insight of this dependency on the load, we examined a base station with $C_0 = 3.2$ Mbps capacity. The VBR nature of the connections was fulfilled by means of three possible transmission rates: 32 kbps, 64 kbps and 128 kbps. An amount of 320kbps of capacity was reserved for handover connections. In such a system a total arrival rate of 15 calls/minute resulted in a highly overloaded system.

We compared the approximate method with computer simulations. The incoming rate of calls initiated in the cell was set to 4 per minute and we increased the arrival rate of handover calls. The measure of accuracy was the cumulative error that is calculated as: $\sum_{m=0}^{C_0} |p_{sim}(m) - p_{app}(m)|$, where $p_{sim}(m)$ and $p_{app}(m)$ are the probability of having m capacity occupied obtained by simulation and the approximate method, respectively.

We observed, that as the load of the base station increased, the error of the approximation increased rapidly, although even in case of very high arrival rate the cumulative error was only about 0.1! . Under light load conditions the error of the approximation affected only the third or fourth decimal value of the occupancy probabilities.

By this confirmation of the accuracy of our method the following results were obtained by the approximate method only, with the knowledge that the examined system is always not heavily loaded, therefore the results have negligible error.

6.2 Effect of the Reserved Capacity and the Base Station Policy

The following results were achieved in a system with 20 Mbps channel capacity. Three types of connections were supposed: voice calls with 32 kbps transmission speed, type 1 multimedia connections with three possible transmission rates (64 kbps lowest rate, 256 kbps average and 512 kbps peak rate) and type 2 multimedia calls (128 kbps lowest, 256 kbps average and 386 kbps peak rate).

To examine the effect of available capacity for different user types we lowered the amount of available capacity for type 1 new calls, while the available capacity for new voice calls was 19.6 Mbps. Figure 1 shows the blocking probabilities of all connection types as the amount of available capacity for insensitive new calls decreases. The left graph shows the results if for all connection types policy 1, i.e. immediate blocking was applied, the right side of the figure shows the case when for type 2 handover calls policy 2 was applied.

The available capacity for blocking insensitive connection is given as the proportion of the total capacity. As we decrease the available capacity for insensitive new calls its blocking probability dramatically increases, however it results in a slight decline of the blocking probabilities of all other types. As it is clear from the graph applying policy 2 for type 2 handover calls results in a 0.01 decline of the blocking probability for this type.

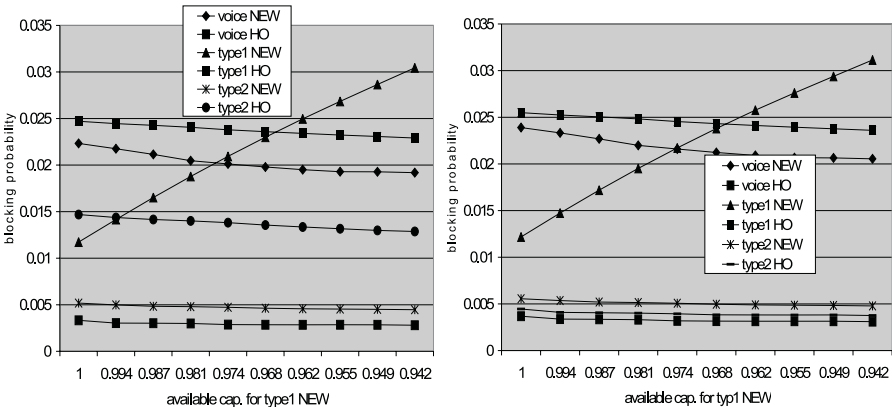


Fig. 1. Blocking probabilities

References

1. M. Ajmone Marsan, S. Marano, C. Mastroianni, M. Meo, Performance Analysis of Cellular Mobile Communication Networks Supporting Multimedia Services, in *Proceedings of MASCOTS'98*, July 19-24, 1998, Montreal, Canada
2. Orlik, P.V.; Rappaport, S.S. Traffic Performance and Mobility Modeling of Cellular Communications with Mixed Platforms and Highly Variable Mobilities. *Proceedings of the IEEE*, Vol. 86, No. 7, July 1998, pp. 1464-1479
3. Orlik, P.V.; Rappaport, S.S. A model for teletraffic performance and channel holding time characterization in wireless cellular communication with general session and dwell time distributions *Selected Areas in Communications*, IEEE Journal on, vol 16, Issue: 5, June 1998, pp. 788 -803
4. Zonoozi, M.M.; Dassanayake, P.; Faulkner, M. Mobility modelling and channel holding time distribution in cellular mobile communication systems. *Global Telecommunications Conference*, 1995. GLOBECOM '95., IEEE, Vol. 1 , 1995 pp. 12 -16
5. A. Bobbio and M. Telek, A benchmark for PH estimation algorithms: results for Acyclic-PH, *Stochastic Models*, vol. 10, pp. 661-677, 1994.
6. M. Neuts. Probability distributions of phase type. In *Liber Amicorum Prof. Emeritus H. Florin*, pages 173-206. University of Louvain, 1975.
7. Sen, P.; Maglaris, B.; Rikli, N.-E.; Anastassiou, D. Models for packet switching of variable-bit-rate video sources. *IEEE Journal on Selected Areas in Communications*, Vol. 7 No. 5, June 1989 pp. 865 -869
8. Special Issue on Packet Speech and Packet Video *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, 1989
9. Kaufman, J. Blocking in a Shared Resource Environment *IEEE Transactions on Communications*, Vol.Com-29, No. 10, Oct. 1981 pp. 1474-1481
10. Roberts, K. *Performance of Data Communication Systems and their Applications*. North-Holland-Elsevier Science Publishers, 1981
11. P. Fazekas, S. Imre, Traffic analysis of broadband cellular networks with multimedia connections, submitted to IFIP ATM & IP 2001, Budapest, Hungary, 2001

Scheduling Disciplines in Cellular Data Services with Probabilistic Location Errors¹

Jiann-Liang Chen, Hao-Chin Cheng, and Han-Chieh Chao

Department of Computer Science & Information Engineering
National Dong Hwa University
Hualien, Taiwan
Lchen@mail.ndhu.edu.tw

Abstract. To provide cellular data services with differentiated *QoS* in a high data error-rate environment, a λ -*WFQ* scheduling discipline, based on the *WFQ* mechanism and LaGrange λ -calculus was developed. The air resources are allocated using λ -calculus and then the *WFQ* mechanism is responsible for the transmission scheduling. λ -*WFQ* discipline compensated for the penalty derived from the location-dependent errors using the equivalent efficiency concept. This discipline can generate a fair schedule for a diverse mix of traffic with diverse *QoS* requirements in a limited radio spectrum. The experimental results show that as much as 5% improvement in the mean acceptance rate is obtained relative to other existing *WFQ-based* schemes at the expense of a small blocking performance.

1 Introduction

Cellular Communications have become the fastest growing segment of the communications industry over the past decade. The objective of a wireless cellular data communications is to provide to users radio access to services comparable to those currently offered by the fixed infrastructure, resulting in a seamless convergence of both fixed and mobile services [1,2]. The introduction of 3G broadband properties has contributed to the increase in cellular data communications [3]. Multimedia applications are expected to become widespread on 3G systems. Because air interfaces create a variety of problems due to interferences, Quality of Service (*QoS*) issues have appeared [4,5].

To support a diverse mix of traffic with diverse *QoS* requirements in 3G systems, we need a perfect mechanism to schedule the traffic resources to meet the *QoS* requirements. Based on the *WFQ* mechanism and LaGrange λ -Calculus, we exploited a novel scheduling discipline, called λ -*WFQ*, at the transmitter-end to ensure high *QoS* [6,7]. The resources are allocated using λ -Calculus and then the *WFQ* mechanism is responsible for the transmission scheduling. An acceptance indication,

¹ This paper is a partial result of project no. of NSC-89-2213-E-259-01 conducted by the National Dong Hwa University under the sponsorship of National Science Council, ROC.

AI , is referred to as the QoS index. Based on the traffic class and residual resource, the λ -WFQ mechanism can generate a fair and dynamic schedule to guarantee QoS .

While packet scheduling for wired links is a maturing area, the scheduling of wireless cellular data links is not a well-established science or even a stable craft. A fundamental difference between wired and wireless links is that wireless media can exhibit substantial radio link error rates, resulting in significant loss of link capacity. In a W -CDMA simulation, we have shown that by using a successive interference cancellation (SIC) scheme, one can effectively estimate and cancel a co-channel interference and thus substantially reduce the near-far effect [8]. The average Bit-Error-Rate (BER) in an $AWGN$ channel is shown in Figure 1. However, all flows experiencing an error rate do not receive their expected QoS .

To support QoS guarantees for multimedia services in cellular data networks with location-dependent errors, the resource scheduling scheme must take the penalty derived from the errors into considerations. However, the errors are unknown and non-deterministic in advance, the resources will be wasted if too many resources are reserved [9]. Based on the WFQ mechanism, LaGrange λ -calculus, and equivalent efficiency concept, a λ -WFQ scheduling discipline was developed to commit cellular data services with differentiated QoS [10].

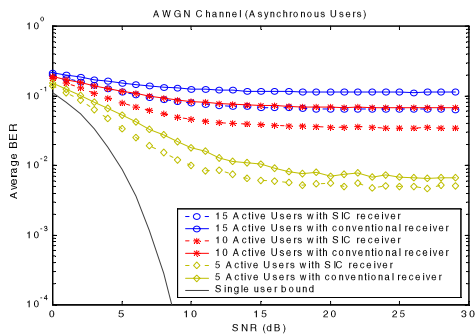


Fig. 1. Comparison of BER with and without SIC in AWGN channel

2 Background Knowledge

The existing architectures, scheduling module, and objective functions relative to this research are described in the following.

2.1 Cellular Service Architecture

Cellular service architecture usually depends on a wireless network. This architecture is a hierarchical structure consisting of a backbone network, mobile switching center (MSC), base stations (BS) and mobile units. The backbone network is a wired network connecting the existing wired links to the MSC or MSC to MSC . The MSC connected to the BS is a special switch tailored for mobile applications. The BS manages the

communications activity of a covered geographic area called a cell where the mobile unit stays. A *BS* is usually in the center of a cell and neighboring cells overlap with each other to ensure the continuity of communication while the mobile users move from one cell to another.

The air interfaces create a variety of problems for mobile units and *BSs* due to interferences. The interference invokes the data transmission errors instantaneously and makes the delay raised. It may no longer be possible for air links to meet the *QoS* commitments. We must decide how much air resources to assign to each active mobile unit. We also must decide how much extra air resources to assign to high-error mobile units at the expense of others.

2.2 WFQ Scheduling Discipline

The fair queuing algorithm, *WFQ*, is a scheduling and multiplexing discipline capable of providing end-to-end delay and bandwidth guarantees on a per connection basis. *WFQ* works by simulating an idealized fluid flow system and serving packets based upon their transmission times in the idealized fluid system. The *WFQ* scheduler distributes air resources according to weights provided by a call scheduling module in a cellular service environment [11].

Recently, a number of *WFQ*-based algorithms based on error-free service model, lead/lag model, compensation model, and slot/packet queue decoupling have been proposed for adapting fair queuing to the wireless domain [6]. The error-free service model provides a reference for how much service a flow should receive in an ideal error-free channel environment. The goal of wireless fair queuing is to approximate the error-free service model by making short error bursts transparent to a flow, and only expose prolonged channel error to the flow. The lagging/leading flow denotes the amount of additional service that the flow must increment/relinquish in the future to compensate for additional service received in the past. The compensation model is to determine how lagging flow makes up their lag and how leading flows give up their lead. In slot/packet queue decoupling, when a packet arrives in the queue of a flow, a corresponding slot is generated in the slot queue of the flow and tagged according to fair queuing algorithm. From the service providers' point of view, we proposed a novel scheduling discipline in which the highest acceptance by all mobile users is achieved to assist the *WFQ* resource scheduling in our research.

2.3 Average Acceptance Rate

When a mobile user wants to communicate with another, a channel must be requested from the base station in the initial stage, which may succeed or fail. If there are available channels, the mobile user will be assigned a channel for communication. If there are no free channels, the request will be rejected. Upon receiving a channel, the mobile user begins to communicate with others and may complete his call at the original cell where it requested a new channel or through other cells. If the mobile user completes his call through other cells, more than two different channels must be used during the call duration (to avoid interference). The procedure for requesting a new channel while the mobile user moves across a cell's boundary is generally called

a handoff. If the handoff succeeds, the mobile user continues his communication without interruption. If the handoff fails, the mobile user's call is force-terminated.

Based on the call life flow, a request and satisfactory service for a mobile user is accomplished with

- A. A request will not be blocked in the initial stage
- B. Service will not be force-terminated in the handoff stage
- C. The quality of service is high during the service hold time

Committing to mobile users' expectations, we defined an Acceptance Indication, AI , as the QoS measurement in cellular data services. Therefore,

$$AI = K_1 (1 - P_b) + K_2 (1 - P_f) + K_3 P_s \quad (1)$$

$$P_b = \sum p(i, j, k) \quad (2)$$

$$P_f = p(C_h, j, k) \mid_{j=C_f - C_h, k=C_d, C=C_f + C_d} \quad (3)$$

$$P_s = \mu [P_c T_c] * (C_{i, Allocate} / C_{i, Max}) \quad (4)$$

$$K_1 + K_2 + K_3 = 1 \quad (5)$$

Where

P_b : the estimated probability that a request will be blocked

P_f : the estimated probability that a service will be force-terminated

P_s : the percentage that indicates the degree of satisfaction during the hold time

P_c : the estimated probability that a call will be completed

C_h : the number of reversed time slots for a handoff call

C_f : the fixed channels of a cell

C_d : the dynamic channels of a cell

$C_{i, Allocate}$: the expected number of time slots allocated to a request for the i th service

$C_{i, Max}$: maximum requirement capacity at a request for the i th service

C : the total available resources at the scheduling time

T_c : the mean service time for a completed call

μ : the mean service rate

K_1, K_2, K_3 : the weighting values that concern mobile users

In the above parameters, P_s indicates the mobile user's satisfaction at the call connection, and $1/\mu$ is the mean service time for a completed call. To increase flexibility, a *Hybrid Channel Allocation* scheme was adopted in our research. In this scheme, the channels are divided into fixed and dynamic sets. The fixed channels, C_f , are assigned to different cells and all of the users share the dynamic channel, C_d .

3 λ -WFQ Discipline

A fairness scheme, λ -WFQ mechanism, based on the LaGrange λ -calculus is proposed for scheduling resources in 3G data services with QoS provisioning. Scheduling issue may be identified by the following function and constraints.

Objective function

$$\text{Max}(AI_T), \quad AI_T = AI_1 + AI_2 + AI_3 + \dots + AI_n \text{ for } n \text{ service request}$$

Subject to

1. $\sum_{j=1}^n C_{j, Allocate} \leq C \Rightarrow \phi \equiv 0 \leq C - \sum_{j=1}^n C_{j, Allocate}$
2. $C_{i, Min} \leq C_{i, Allocate} \leq C_{i, Max} \quad i=1,2,\dots,n$

Where

AI_i : acceptance indication of the i th service (defined on Equation (1))

$C_{i, Min}$: minimum requirement capacity at a request for the i th service

Resource scheduling is a constrained optimization problem that may be attacked formally using advanced calculus methods that involve the LaGrange λ -calculus. To establish the necessary conditions for an extreme AI_T value, add the constraint function to the AI_T after $\emptyset$ has been multiplied by a multiplier, λ .

$$\zeta = AI_T + \lambda \emptyset$$

The necessary conditions for an extreme AI_T value results when we take the first derivative of the λ -calculus with respect to each of the independent values and set the derivatives equal to zero. Based on the 3G application domain, we only take the derivative of the λ -calculus with respect to the $C_{i, Allocate}$ values at a scheduling time give the set of equations

$$\frac{\partial \zeta}{\partial C_{i, Allocate}} = \frac{\partial AI_i}{\partial C_{i, Allocate}} - \lambda = 0 \quad i=1,2,\dots,n \quad (6)$$

$$\lambda = \frac{\partial AI_1}{\partial C_{1, Allocate}} = \frac{\partial AI_2}{\partial C_{2, Allocate}} = \dots \quad (7)$$

The necessary condition for the existence of a maximum AI_T for cellular data services is that the incremental AI of all of the mobile users is equal to λ . Of course, to this necessary condition we must add the constraint equation that the sum of the $C_{i, Allocate}$ values must be less than or equal to C . In addition, there are two inequalities that must be satisfied for each request. That is, the $C_{i, Allocate} \geq C_{i, Min}$ and $C_{i, Allocate} \leq C_{i, Max}$. The operating flow chart is shown in Figure 2. The operating scenarios are described as follows.

I. Estimation of AI for each type of service

Many treatments was performed by an OpNet simulator according to the variation of $C_{i, Allocate}$, thus a set of AI values for the specified service class maybe estimated. Through the curve fitting technique for operating data filtering, the AI function for each service class may be treated for the linear or quadratic rate case. Such as

$$AI_i = a_i + b_i C_{i, Allocate} + c_i C_{i, Allocate}^2 \quad a, b, c = \text{constant}$$

II. λ -calculus

Based on the Step I, assuming two types of services whose incremental AI values are represented by the following equations.

$$\frac{dAI_1}{dC_{1, Allocate}} = 2c_1 C_{1, Allocate} + b_1 = \lambda$$

$$\frac{dAI_2}{dC_{2, Allocate}} = 2c_2 C_{2, Allocate} + b_2 = \lambda$$

$$Cr = \min\{C, C_{1, Max} + C_{2, Max}\}$$

Therefore, $C_{1, Allocate}$ and $C_{2, Allocate}$ may be estimated while a λ value is selected.

III. Generation of schedule

The iterative process finding λ value was stopped while the $\sum_{i=1}^n C_{i, Allocate} = Cr$.

Two cases are discussed in the generating schedule.

Case 1: $Cr = (C_{1, Max} + C_{2, Max})$, thus

$$C_{1, Allocate} = C_{1, Max}$$

$$C_{2, Allocate} = C_{2, Max}$$

Case 2: $Cr = C$, thus we recognize the inequality constraints, then the necessary conditions may be expressed as shown in the set of equations making up Equation (8)-(10). Three situations are identified.

$$\frac{dAI_i}{dC_{i, Allocate}} = \lambda \quad \text{for } C_{i, Min} < C_{i, Allocate} < C_{i, Max} \quad (8)$$

$$\frac{dAI_i}{dC_{i, Allocate}} \leq \lambda \quad \text{for } C_{i, Allocate} = C_{i, Max} \quad (9)$$

$$\frac{dAI_i}{dC_{i, Allocate}} \geq \lambda \quad \text{for } C_{i, Allocate} = C_{i, Min} \text{ or } 0 \text{ (Blocked)} \quad (10)$$

Situation I: $\lambda = \lambda_2$ or λ_3 , both of $C_{1, Allocate}$ and $C_{2, Allocate}$ values will fall into the range $\{C_{i, Min}, C_{i, Max}\}$

Situation II: $\lambda = \lambda_1$, the $C_{1, Allocate}$ value either $C_{1, Min}$ or 0 (blocked) because of

$$\frac{dAI_1}{dC_{1, Allocate}} \geq \lambda_1$$

Situation III: $\lambda = \lambda_4$, the $C_{2, Allocate}$ value is equal to $C_{2, Max}$ because of $\frac{dAI_2}{dC_{2, Allocate}} \leq \lambda_4$

IV. Error compensation

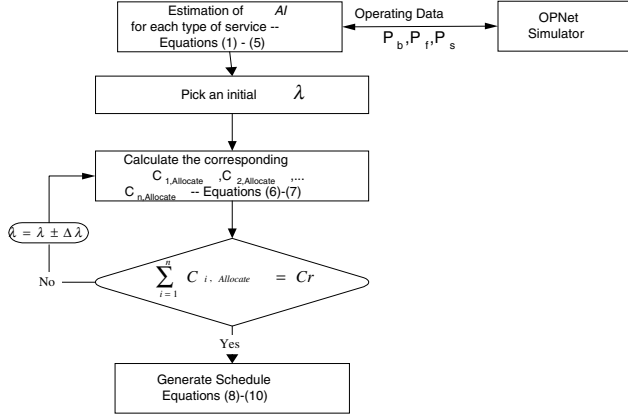
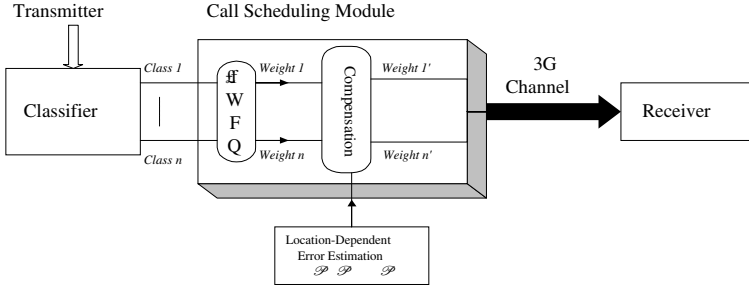
λ -WFQ scheduling discipline extends WFQ algorithm via dynamic weight adjustments to compensate for the penalty derived from location-dependent errors and improve the service quality. The weight values are tuned based on the equivalent efficiency concept. The scheduling model is shown in Figure 3. The compensation scenarios are described in the following.

Error Free Environment

The cellular data services are invoked in an error free environment. That is, these services will actually obtain the reserved resources scheduled by the λ -WFQ scheme.

Location-Dependent Error Environment

To support QoS guarantees for multimedia services in cellular data networks with location-dependent errors, we must decide how extra resources are assigned to high-error mobile units such that their QoS is supported.


 Fig. 2. The flow chart of λ -calculus

 Fig. 3. λ -WFQ scheduling via error compensation

To achieve the high AI, an equivalent efficiency concept is proposed here. We defined a penalty factor $\mathfrak{S}$ to compensate for the effect of location errors. The functional block is illustrated in Figure 4.

$$\mathfrak{S} = \frac{n}{\frac{1}{1-P_1} + \frac{1}{1-P_2} + \dots + \frac{1}{1-P_n}} \quad (11)$$

where $P_1, P_2, \dots, P_n$ are the location error probabilities of 1th, 2th, ..., and nth service.

Thus the new weight value of i th service is,

$$\text{Weight } i' = \text{Weight } i * \mathfrak{S} * 1/(1-P_i) \quad (12)$$

However, the $R_{res-i} = B * \text{Weight } i'$ for several guaranteed QoS services may be not enough. Hence, the reserved resources are transferred from best effort services to guaranteed QoS services.

- *Guaranteed QoS services*

$$\text{Add } R_{res-i} = \frac{\{B * \text{Weight } i\}}{P_i} - \{B * \text{Weight } i'\} \quad (13)$$

• *Best-effort services*

.Subtract
$$R_{res-j} = \frac{R_{res-j}}{\sum_{j \rightarrow \text{best service}} R_{res-j}} * \sum_{i \rightarrow \text{QoS guarantee service}} (Add_{ee} R_{res-i}) \quad (14)$$

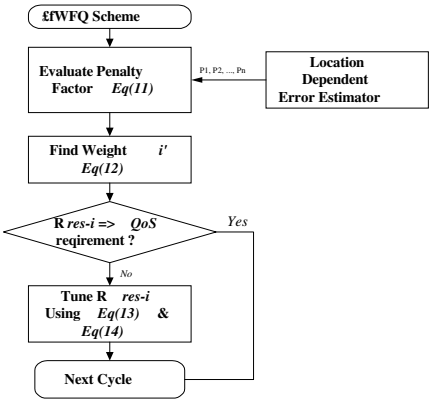


Fig. 4. Compensation scenarios

4 Performance Analysis

An investigation will be done using Notebooks with *Ericsson S888* handsets and *Nokia Card Phone* in the Telecommunications cellular system, Taiwan. The datagram is transmitted from the source end of experimental channels (1 control time slot + 7 traffic time slots) through PSTN network to the destination end of experimental channels. An example is shown as follows. Three types of services including voice phone, file transfer, and interactive video, are requested at the scheduled time. The parameters listed in Equations (2), (3), and (4) were measured and evaluated using an OpNet simulator. The relationship between AI_i and $C_{i,Allocate}$ are identified. Through the curve fitting techniques, the AI_i functions are fixed and shown in Figure 5.

Based on the individual AI functions of each request, we observed the following simulation results.

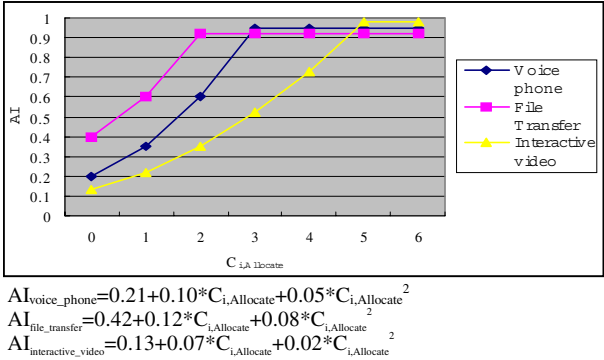


Fig. 5. The AI function for each request ($K_1=0.2, K_2=0.4, K_3=0.4$)

1. **Fair Resource Sharing:** If the minimum and maximum resources requested by the three requests were (0,3), (0,2), and (0,5), respectively, and the available resource at the scheduled time was from 1 to 8 time units, the simulation result is as shown in Figure 6. In Figure 6, we observe that many time units are allocated to interactive video and voice phone. A mobile user expects a higher quality for real-time services. The more the available resources, the higher the rate.
2. **Over-Rating Resource Sharing:** The incremental rate is over λ when $C_{i,Allocate} > C_{i,Max}$. In this situation, Equation (9) is suitable to resolve the problem. In the simulation, this situation occurs if the resource is greater than 8 time units.
3. **Under-Rating Resource Sharing:** In the simulation, all $C_{i,Min}$ values were set to zero, thus the situation that the incremental rate would be under λ did not occur. However, the $C_{i,Min}$ value in real-time services is always not equal to zero, thus Equation (10) is suitable to resolve the problem. In the simulation, this situation occurs if the required resource of voice phone is modified to (1,3) and the available resource is less than 2 time units.
4. **Network Efficiency:** The network efficiency is dependent on the service throughput. Two video-based services were invoked in different error environments. The efficiency analysis is shown in Figure 7. It was found that the efficiency is less than 50% if error rate of one of the services is greater than 0.5.
5. **Performance Analysis:** Four services were invoked in different error environments. Four cases were tested and the results are listed in Table 1. Different capability B may affect the AI values. However, different K values may generate different results. From the simulation results, we found that the average AI values were dominated by the QoS guarantee services. In addition, the greater the residual capability is, the higher the average AI is. The results show that the average AI of the schedule derived from our approach provides 5% higher values than the traditional WFQ scheme.

5 Conclusion

A fairness scheme based on the WFQ mechanism, optimization theory, LaGrange λ -calculus and equivalent efficiency concept was proposed for shared resources in cellular data services with QoS provisioning and location-dependent errors. The simulation results for three traffic classes clearly demonstrate higher acceptance with mobile users at the expense of a small blocking performance. Similar methodologies can be applied for large-scale cellular data service systems with differentiated QoS .

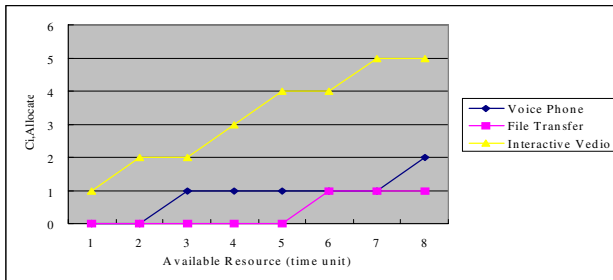


Fig. 6. The simulation result

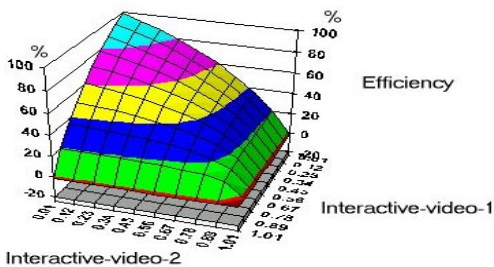


Fig. 7. Network efficiency for two video-based services

Table 1. Case study ($B=30A$, $K_1=0.1, K_2=0.2, K_3=0.7$)

	Case 1			Case 2			Case 3			Case 4		
	<i>pi</i>	<i>Rres</i>	<i>output</i>	<i>pi</i>	<i>Rres</i>	<i>output</i>	<i>pi</i>	<i>Rres</i>	<i>output</i>	<i>pi</i>	<i>Rres</i>	<i>output</i>
Voice Phone	0	1.0A	1.0A	0	1.0A	1.0A	10	1.1A	1.0A	20	1.2A	1.00A
Interactive video	0	8.0A	8.0A	30	11.4A	8.0A	40	13.3A	8.0A	0	8.0A	8.00A
File transfer 1	0	10.5A	10.5A	0	7.2A	7.2A	15	8.0A	6.8A	30	8.6A	6.05A
File transfer 2	0	10.5A	10.5A	30	10.3A	7.2A	10	7.5A	6.8A	50	12.1A	6.05A
Total		30.0A	30.0A		30.0A	23.4A		30.0A	22.6A		30.0A	21.10A
Average AI		0.73			0.68*/0.71			0.59*/0.68			0.65*/0.73	

*results from traditional *WFQ* scheme

References

1. Ivano, G., Paolo, D., Paolo, F.: The Role of Internet Technology in Future Mobile Data Systems. *IEEE Communications Magazine*. Vol.38. No.11. (2000) 68-73

2. Lin, Y.B., Chlamtac, I.: *Wireless and Mobile Network Architectures*. Wiley, (2000)

3. Huber, J.F., Weiler, D., Brand, H.: UMTS, The Mobile Multimedia Vision for IMT 2000: a Focus on Standardization. *IEEE Communications Magazine*. Vol.38. (2000) 129-136

4. Mahadevan, I., Sivalingam, K.M.: Quality of Service Architectures for Wireless Networks: IntServ and DiffServ Models. *Proc. Workshop on Mobile Computing*. (1999) 414-425

5. Choi, S., Shin, K.G.: An Uplink CDMA System Architecture with Diverse QoS Guarantees for Heterogeneous Traffic. *IEEE/ACM Transactions on Networking*. Vol.7. No.5. (1999) 616-628

6. Bharghavan, V., Songwu, L., Nandagopal, T.: Fair Queuing in Wireless Networks: Issues and Approaches. *IEEE Personal Communications*. Vol.6. No.1. (1999) 44-53

7. Chen, J.L., Chao, H.C.: Resource Sharing Scheme for Cellular Data Services with Different QoS. *IEICE Transactions on Communications*. Vol.E83-B. (2000) 2545-2549

8. Chen, J.L., Yang, M.H., Cheng, P.Y.: An Agent-Based Scheme for Interference Cancellation in a DS/CDMA System. *Proc. IEEE Conference on System, Man, and Cybernetic*. (2000) 437-442

9. Chen, J.L., Kung, J.H., Chao, H.C.: Resource Reservation with Mobile Hosts Using Fuzzy Matrices. *International Journal of Computer Communications (COMCOM 1832)*: 2001

10. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Service. *RFC2475*. (1998)

11. Simon, R., Sood, A., Mundur, P.: Network Service Selection for Distributed Multimedia Applications. *Proc.Third International Conference on Computational Intelligence and Multimedia Applications*. (1999) 388-392

Restoration from Multiple Faults in WDM Networks without Wavelength Conversion

Chuan-Ching Sue and Sy-Yen Kuo

Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
sykuo@cc.ee.ntu.edu.tw

Abstract. This paper addresses the problem of achieving full and fast restoration to tolerate as many faults as possible in wavelength-routed wavelength division multiplexing (WDM) networks with no wavelength conversion. We model the problem of finding the maximum number of faults tolerated as a constrained ring cover problem, which is a decomposition problem of exponential complexity. Three heuristic methods which guarantee that at least one fault can be tolerated are proposed. The Ear Decomposition (ED) method can always generate a decomposition to guarantee that only one fault can be tolerated. The Planar Decomposition (PD) method, which takes advantage of the bipartite graph model to generate a decomposition, can tolerate up to f faults, where f is the maximum cardinality between the two bipartite vertex sets. The Maximally Separated Rings (MSR) method uses the greedy method to find a decomposition to tolerate as many faults as possible. The marked-link (ML) method is also proposed to enhance the performance by marking some links, which are originally used for protection, available for normal transmissions. Finally, we also evaluate the number of faults tolerated and the blocking probabilities of these methods in three example networks.

1 Introduction

As the WDM technology is introduced into networking, survivable and flexible optical networks will be deployed widely in the near future. In particular, restoration of services within a short period of time frame after a link or node failure is becoming a requirement.

Several protection and restoration schemes for enhancing network reliability have been presented in [1-5, 7, 10]. If we have a fault detection capability in the OXC as in [9], fault detection can be done only at the endpoints of where the failure occurs. The fault recovery in the path/end-to-end mode cannot be completed within the optical layer and must be left to the upper layer since it needs acknowledgement. While the link restoration mode is based on the localized control and the coordination can be completed within the optical layer if the rerouting paths are static and the resources are sufficient. The proposed restoration scheme is thus based on localized control and can be executed within a sub-second time frame instead of relying on an online computation of backup routes which is slower but more bandwidth efficient.

The number of wavelength convertible nodes in the network is another factor which has a significant impact on the degree of localization and coordination needed for restoration. This has been demonstrated in a pure ring, in which the lack of wavelength conversion greatly simplifies and expedites the activation of restoration [4]. Unfortunately, in a mesh network with only one bi-directional fiber pair in each link, channel protection is inherently dependent on wavelength conversion since the assignment of working and protection channels can conflict with each other, and thus it is not directly applicable to a network without wavelength conversion. Further, the number of faults tolerated is only one in the previous works for either link fault or node fault model [1, 10]. Little work has been done to increase the number of faults tolerated except the study in [7], while it required the network topology to be planar.

Therefore, in this paper we consider WDM networks without wavelength conversion. The main objectives of the proposed scheme are fast and full restoration while tolerating as many multiple faults as possible. Our method creates two fully connected directed sub-graphs (Working (WSN) and Protection (PSN)) of a network by three heuristic methods such that after some edge/node failures all operational nodes are still connected by the WSN or PSN sub-graph. We model the problem of achieving the maximum number of faults tolerated in the construction as a constrained ring cover problem and show it to be NP-complete. Three heuristic methods are Ear Decomposition (ED), Planar Decomposition (PD), and Maximally Separated Rings (MSR). The ED method can always generate a decomposition to guarantee that only one fault can be tolerated. The Planar Decomposition (PD) method, which model the problem as a bipartite graph, can generate a decomposition to tolerate up to f faults, where f is the maximum cardinality between the two bipartite vertex sets. The Maximally Separated Rings (MSR) method uses the greedy method to find a decomposition to tolerate as many faults as possible.

The rest of this paper is organized as follows. In Section 2, we give the preliminaries for our restoration scheme. In Section 3, the problem of finding a decomposition to tolerate maximum number of faults is modeled as a ring cover problem and three heuristic methods for this problem are detailed. Section 4 depicts the performance evaluation. Finally, we conclude our study in Section 5.

2 Preliminaries

In this section, we introduce the basic idea that how a fast and full restoration can be achieved by simple loop-back mechanism under various fault scenarios.

2.1 Single Fault

The algorithm to selecting proper directions that satisfy the full restoration conditions for link/node-redundant networks was introduced in [1]. The algorithm is based on the ear decomposition (ED) method, which is a technique of decomposing a given network into simpler parts such that the computation on the simpler parts corresponds to the computation on the entire network. The time complexity of the algorithm is $O(V+E)$ since it is based on the depth first search. Based on *st-numbering* [14], we further exploit a scheme of *st-looping* to find the associated ring cover set for

explaining that the ear decomposition method can only tolerate one fault. An example is shown in Fig. 1 to see the results of the algorithm. Fig. 1(a) is an example 16-node graph whose nodes meet the *st-numbering* rules if $(s, t) = (1, 16)$. Fig. 1(b) and Fig. 1(c) show the results of selecting the directions in all the links.

For the graph shown in Fig. 1, the associated ring cover sets are $R_1=\{1, 8, 9, 10, 11, 13, 14, 16\}$, $R_2=\{1, 2, 3, 7, 10, 11, 13, 14, 16\}$, $R_3=\{3, 4, 6, 15, 16, 1, 2\}$, $R_4=\{1, 5, 6, 15, 16\}$, and $R_5=\{6, 12, 13, 14, 16, 1, 5, 6\}$.

An interesting observation is that all rings in the associated ring cover set contains the edge $(1, 16)$, that is, (s, t) . This makes the method be able to tolerate one fault only under the full restoration requirement since the number of backup wavelengths can be less than the number of affected working wavelengths in the link (s, t) . Thus, using the method based on the ear decomposition can only tolerate one fault.

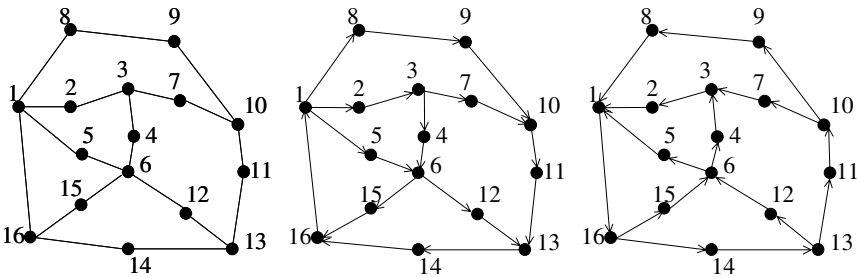


Fig. 1. (a) Example *st*-numbered graph, (b) sub-graph numbered increasingly except (s, t) , (c) sub-graph numbered decreasingly except (s, t) .

2.2 Multiple Faults

We consider an example network and its possible double cycle covers (DCC). On the basis of these double cycle covers, we discuss whether double cycle covers can be used in a WDM network without wavelength conversion to survive multiple faults. Fig. 2 shows a possible double cycle cover in bold lines for the example topology in Fig. 1.

For simplicity, let us consider just two wavelengths A and B. Fig. 3 shows that we cannot use one ring to protect another unless we have wavelength convertible nodes at nodes 1, 3, 10, and 16, and thus incur significant network management overhead. Therefore, using the double cycle cover for WDM recovery would require wavelength conversion.

The maximum number of link failure that can be recovered is $\lfloor f/2 \rfloor$, where f is the number of faces. In the worst case, for a bi-directional ring with inner and outer face, only one fault can be restored. Although the number of faults tolerated is greater than one, the DCC can not be applied to a WDM network without wavelength conversion. In the following section, we will find a decomposition method to allow multiple faults without the need of wavelength conversion. Note that “multiple faults” in this paper means multiple link faults that are not introduced by a single faulty node.

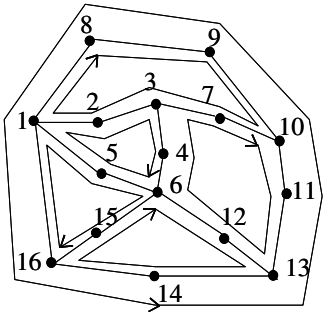


Fig. 2. A double cycle cover for a 16-node network.

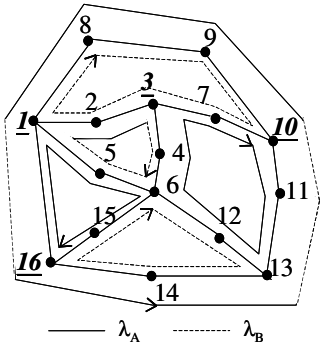


Fig. 3. Double ring cover does not work without wavelength converters.

3 Problem & Heuristic Methods

After the above discussions, we know that the ED method can work in a WDM network without wavelength conversion, but not for multiple faults. In contrast, the DCC works for multiple faults but fails on WDM networks without wavelength conversion. A comprise can be made to relax the ring cover set from a double cycle cover to a cycle cover without conflict directions in each link. Once such ring cover set is found, the direction in each link is accordingly defined and the graph can be decomposed into two connected directed sub-graphs. This is similar to ED except that each member in the ring cover set does not always meet the special edge (s, t) . Thus, tolerating multiple faults is possible for a network with such ring cover set.

Given a graph $G = (V, E)$, our goal is to find a best ring cover set C for G to tolerate the maximum number of link faults. This problem is difficult computationally. Since no polynomial time algorithm exists to solve the above problem, heuristic methods are necessary. The first heuristic method follows the ED method to guarantee that the maximum number of faults tolerated is one. The second one is called the planar decomposition (PD) heuristic because it is based on the planar property. It denotes each possible cycle by a new vertex, and determine whether the edge between each vertex pair exists or not through the intersection of the cycles.

That is, when an edge exists between two vertexes, it means the two cycles have some intersections.

If the graph formed by the new model can be two-colorable, it means that the direction assignment in all links is done. Two-colorable graphs are also called bipartite graphs. The maximum cardinality between the two parts of the bipartite graph is the maximum number of faults tolerated because they can be restored simultaneously and routed through disjoint paths without blocking one another. However, the graph formed by the new model is not always directly two-colorable. Some cycles (vertices) have to be deleted to make two-colorability possible. In order to keep as many vertices as possible to increase the number of tolerated faults, the deletion must be minimized. This problem of minimum vertex deletion to obtain connected sub-graph with bipartite property has been shown to be NP-complete in [11]. Therefore, the PD method is to find an approximation for the original minimum deletion problem. The third heuristic method MSR (maximally separated rings) is to directly apply the depth first search on the original graph to greedily locate rings in the network. In fact, it looks for rings iteratively. That is, this heuristic method tries to avoid covering a link with many different rings. The procedure continues until every link is covered by some ring in the cover.

We give an example illustration in Fig. 4 for the PD heuristic method.

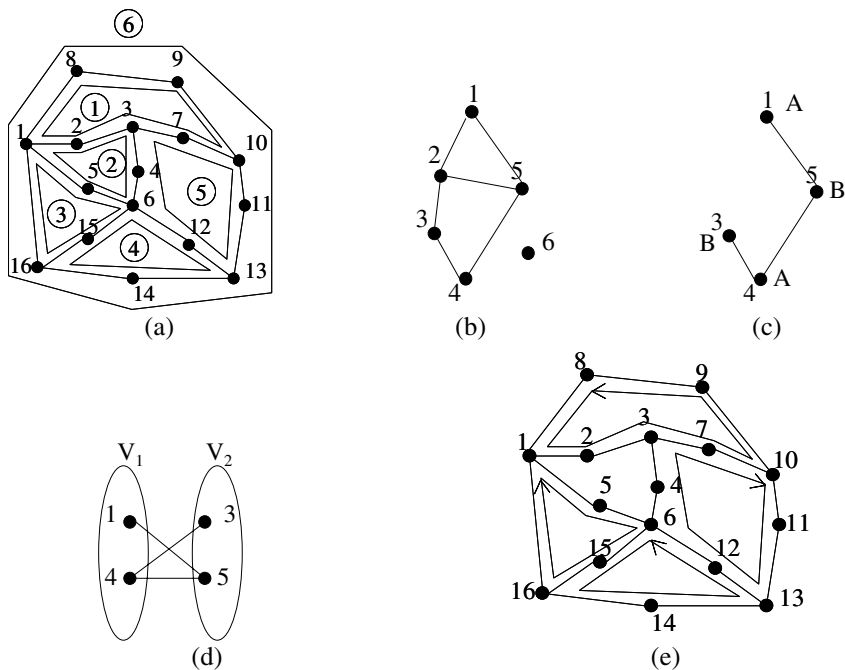


Fig. 4. The process of the PD heuristic on the example network in Fig. 1.

Based on the same example network in Fig. 1, the first step is to find the basis of the cycle cover set. The possible planar faces are shown in Fig. 4(a). There are six possible faces in the original network. Since the links of the outer face completely contains the inner faces, the induced graph is shown in Fig. 4(b), which is still

connected without vertex 6. The next step is to delete the vertices to make the graph two-colorable. In Fig. 4(c), vertex 2 is removed from the resulting graph. The final two-colorable graph must be a bipartite graph as shown in Fig. 4(d). The maximum cardinality of the two set V_1 and V_2 is two. Therefore, the decomposition derived by the heuristic can survive two faults and the final decomposition is shown in Fig. 4(e).

All these heuristic methods are applied directly to the original network topology. We observed that some links can be removed while maintaining the network connectivity. This corresponds to the minimal sub-graph with edge deletion, which is also an NP-complete problem. The removed links are marked. The marked links can only be used to respond to the needs of the network. Although the average length of restoration paths can be longer when the marked links are used, the restorability is not reduced and the blocking probability is smaller in working channels. We call our enhancement method as the marked-links (ML) method, which greedily find the available removable links. The following section is to evaluate the blocking probability loss for the full restoration and the blocking probability gain with the ML method.

4 Performance Evaluation

In order to formally compare the performance of the proposed heuristic methods with that of the unprotected method, we start by making the standard series independent link assumption introduced by Lee and commonly used in the analysis of circuit-switched network [6]. Let p be the probability that a wavelength is used on a hop and F be the available wavelength on a fiber. Note that since pF is the expected number of busy wavelengths, p is a measure of the fiber utilization along this path. Now consider a network without wavelength conversion. The probability of blocking P_b is the probability that each wavelength is used at least in one of the H hops, i.e.,

$$P_b=[1-(1-p)^H]^F \tag{1}$$

Three example networks, National Network, Icosahedron, ArpaNet, are shown in Fig. 5 and used to evaluate the number of faults tolerated and the blocking probability. The main topology features of these networks and the experimental results are presented in Table 1.

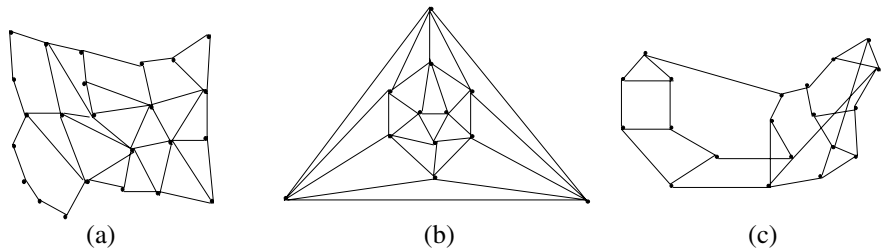


Fig. 5. (a) National network. (b) Icosahedron network. (c) ArpaNet network.

Note that V is the number of nodes, E the number of links, δ the average node degree, H the average hop distance for shortest path routing on all node pairs, and the

three heuristic methods are: ED, PD and MSR. Further, the National network and Icosahedron network are planar, while the ArpaNet is non-planar. Note that when performing PD on ArpaNet, two links are ignored to keep planarity.

In general, the PD and the MSR can always find a better ring cover set to select the directions for the links of a network than ED. In the following, we focus on the blocking probability for three heuristic methods and compare them with the unprotected method using the formula derived in Equation (1).

Table 1. Main topological parameters of the example networks and the number of faults tolerated.

	Parameters				# of faults tolerated		
Networks	V	E	δ	H	ED	PD	MSR
National	24	41	3.38	2.93	1	9	9
Icosahedron	12	29	5	1.64	1	7	7
ArpaNet	20	31	3.01	2.81	1	3	3

From Fig. 6, it is observed that the three heuristic methods have no significant differences in the blocking probability. When the ML method is considered, the change is more than the difference between three heuristic methods. In addition, after more marked links are introduced in the working sub-graph, the blocking probability can be improved as much as the increase in the number of marked links. However, the unprotected method always outperforms the three heuristic methods and the marked-link method since it dose not waste the resources for the purpose of protection.

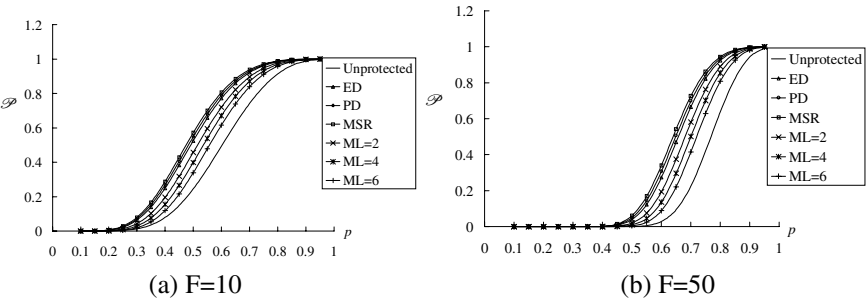


Fig. 6. The blocking probabilities for the three heuristic methods, the unprotected method, and the ML method for the National network.

Another problem is to find the worst case such that the load can make the difference larger. In Fig. 7, it clearly shows the differences in blocking probability between the three heuristic methods, the marked-link method, and the unprotected method for various loads. It has a saturation load that makes the difference the worst. In Fig. 7(a) with $F=10$, the saturation occurs at $p=0.55$, while in Fig. 7(b) with $F=50$, the saturation occurs at $p=0.7$. It can be concluded that when the number of wavelengths increases, the load that will cause saturation is higher. Combining the

results of Fig. 6 and 7, we can operate a network at a load lower than the saturation point to keep the differences in blocking probability low and to keep the absolute value of blocking probability low, too.

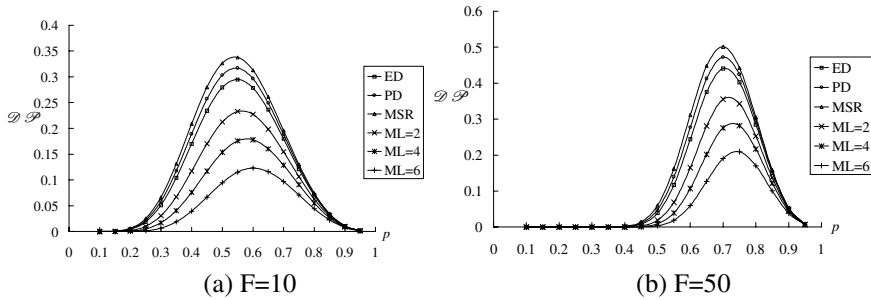


Fig. 7. Differences in blocking probability between the three heuristic methods, the marked-link method, and the unprotected method.

5 Conclusions

This paper proposed a fast and full restoration mechanism under multiple-faults scenario for WDM networks without wavelength conversion to alleviate the management overhead. Three heuristic methods: Ear Decomposition (ED), Planar Decomposition (PD) and Maximally Separated Rings (MSR), are proposed to find proper directions in the ring cover set. The Marked-Link (ML) method is proposed to further improve the blocking probabilities of working channels. The analytical results also show that we can operate a network below a certain saturation load to make the difference of blocking probability smaller between the proposed heuristic methods and the unprotected one while keeping a lower absolute blocking probability.

References

1. M. Medard, S. G. Finn, and R. A. Barry, "WDM Loop-back Recovery in Mesh Networks," *Proc. Infocom '99*, pp. 752-759, Mar 1999.
2. S. Ramamurthy and B. Mukherjee, "Survivable WDM Mesh Networks, Part I – Protection," *Proc. Infocom '99*, pp. 744-751, Mar 1999.
3. Y. Miyao and H. Saito, "Optimal Design and Evaluation of Survivable WDM Transport Networks," *IEEE J. Select. Areas Commun.*, vol. 16, no. 7, pp. 1190-1198, Sep. 1998.
4. O. Gerstel, et al., "Fault Tolerant Multiwavelength Optical Rings with Limited Wavelength Conversion," *IEEE J. Select. Areas Commun.*, vol. 16, no. 7, pp. 1166-1178, Sep. 1998.

5. S. Baroni, et al., "Link failure restoration in WDM Optical Transport Networks and The Effect of Wavelength Conversion," *Proc. Int'l Conf. Optical Fiber Communication*, pp. 123-124, Feb 1997.
6. R. Barry and P. Humblet, "Models of *Blocking* Probability in All-Optical Networks with and Without Wavelength Changers," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 858-866, June 1996.
7. G. Ellinas and T. Stern, "Automatic Protection Switching for Link Failures in Optical Networks with Bi-directional Links," *Proc. IEEE Int'l Conf. Globecom*, pp. 152-156, Nov. 1996.
8. S. Even, "Graph Algorithms," Computer Science Press, 1979.
9. C. Li and R. Ramaswami, "Automatic Fault Detection, Isolation, and Recovery in Transparent All-Optical Networks," *IEEE J. Lightwave Technol.*, vol. 15, no. 10, pp. 1784-1793, Oct. 1997.
10. S. G. Finn, M. Medard, R. A. Barry, "A New algorithm for bi-directional link self-healing for arbitrary redundant networks," *Proc. Int'l Conf. Optical Fiber Communication*, pp. 298-299, Feb 1998.
11. Yannakakis, M., "The effect of a connectivity requirement on the complexity of maximum subgraph problems," *J. ACM*, pp. 618-630, May 1979.

An All-Optical WDM Packet-Switched Network Architecture with Support for Group Communication

Marcos Rogério Salvador, Sonia Heemstra de Groot, and Diptish Dey

Centre for Telematics and Information Technology (CTIT), University of Twente

P.O. Box 217, 7500 AE, Enschede, The Netherlands

Tel +31 53 4898013, Fax +31 534894524

{[salvador](mailto:salvador@cs.utwente.nl), [heemstra](mailto:heemstra@cs.utwente.nl), [dey](mailto:dey@cs.utwente.nl)}@cs.utwente.nl

Abstract. The so-called media convergence to the Internet is foreseen. As a consequence of this convergence, MANs will face new demands not only in terms of bandwidth, but also in terms of services. To meet these new demands, new MAN architectures are required. WDM-based MAN architectures that tackle the first problem are available not only in the literature, but also on the market. In this paper we deal with the second problem. Specifically, we describe a novel WDM-based MAN architecture that supports group communication, a service that is expected to increase considerably as new applications converge to the Internet. Based on packet switching, the architecture supports both point-to-point and point-to-multipoint communication in the optical domain.

1 Introduction

The convergence of various media, applications and networks to the Internet in the near future is foreseen. Such a convergence will affect metropolitan area networks (MANs) directly. Firstly, the demand for bandwidth is expected to increase enormously. It is arguable whether MAN architectures based on electronic technologies will cope with such a foreseen demand. Secondly, Internet traffic will have characteristics that will differ even more from those that drove the design of MAN architectures currently in operation. MAN architectures were mostly designed to cope with long-lived voice flows and, therefore, are circuit-switched. Internet Protocol (IP) traffic, however, is inherently bursty and consists mostly of short-lived data flows and small packet sizes.

The bandwidth demand problem has been tackled with the deployment of wavelength division multiplexing (WDM). WDM is an optical technology that exploits the frequency spectrum of the light, thus enabling several distinct optical channels within a single optical fiber, each carrying up to tens of Gbps.

MAN architectures based on WDM are a fact today. However, they are still based on circuit switching and the store-and-forward (i.e., multihop) communication paradigm. To cope with the characteristics of the next generation Internet, MAN architectures should pursue all-optical (i.e., single-hop) packet switching. Besides the robustness and better resource utilization that are typical of packet switching, all-

optical packet switching eliminates queuing delays at intermediate nodes and provides bit rate and protocol transparencies.

All-optical WDM packet-switched network architectures are described in [1], [2]. Common to these architectures is the fact that they rely on the slotted-ring concept and the use of tunable transmitter (TTx) and fixed receiver (FRx) node architectures. In such networks, the total capacity of each channel is divided into (time) slots of fixed length. Upon arrival of an empty slot, a node tunes its laser onto that slot's wavelength and transmits. At the destination, the payload is obtained from the slot and the slot is released, being reused by either the node itself or downstream nodes.

Whilst TTx/FRx node architectures provide very high performances, they have some drawbacks. Firstly, tunable lasers are too expensive and very difficult to control. Secondly, the use of a single FRx per node makes the support of group communication somewhat inefficient and the demands for group communication are expected to increase considerably as a result of the foreseen convergence. For instance, a source node may have to transmit a given information W times, where W denotes the number of distinct wavelengths that the destination nodes can receive from. This is a considerable drawback.

Dey et al., in [3], propose a node architecture that deals with these problems. Unlike other node architectures, the node architecture relies on a fixed transmitter (FTx), array of receivers (ARxs) configuration. The main benefits of this node architecture are: i) management complexity due to TTxs is eliminated, ii) cheap cost of FTxs and iii) support of group communication is more efficient.

The node architecture has different characteristics and requirements and, hence, requires medium access control (MAC) mechanisms specifically designed for it. We discuss such mechanisms in this paper. We also discuss some performance results that were obtained via simulation activities.

The rest of this paper is organized as follows. In Sect. 2 the network and the node architectures are described. In Sect. 3 the MAC protocol that has been designed for the network and the node architectures described in Sect. 3 is described. In Sect. 4 some performance results are shown and analyzed. In Sect. 5 we conclude the paper.

2 The Network and the Node Architectures

The network architecture is based on an adaptation of the slotted-ring concept to the multi-channel nature of WDM. In this architecture, W wavelength channels are used to carry payload information. A single extra wavelength channel is used to carry control information. The total bandwidth of each channel, including the control one, is divided into (time) slots of fixed length.

Slots across the W payload wavelength channels, herein called payload slots, are synchronized in parallel so as to reach each node all at the same time. Slots on the control wavelength channel, herein called control slots, are sent slightly ahead of their corresponding payload slots. This is to account for the configuration time of the fabrics at the nodes. Fig. 1 illustrates how payload slots and control slots are synchronized.

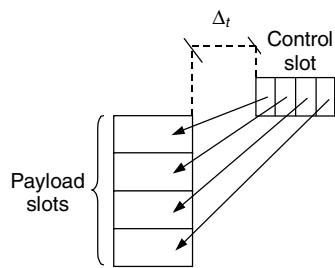


Fig. 1. Slot alignment

The network is partitioned into $S = N / W$ segments¹, where N is the number of nodes in the network. If $N \leq W$ then each node is assigned (via management operation) an exclusive transmission wavelength channel. Otherwise, S nodes, each on a different segment, share the same transmission wavelength. A node can transmit on only one wavelength. On the other hand, a node can receive on all wavelengths simultaneously. Fig. 2 illustrates the basic network architecture. Although out of the scope of this paper, scalability can also be achieved via an interconnected ring structure.

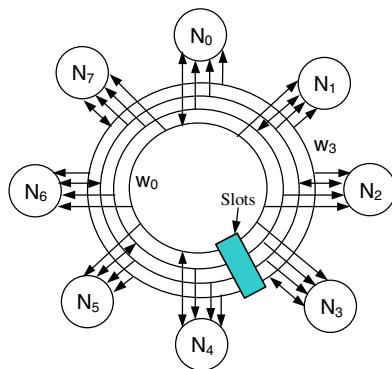


Fig. 2. Example of Network Architecture; $F = 1$ optical fiber; $N = 8$; $W = 4$.

The node architecture is shown in Fig. 3. Each node is equipped with one FTx and an array of W FRxs, each tuned on a distinct wavelength, for payload transmission purposes. Each node is also equipped with one FTx and one FRx, both operating on the same wavelength, for control information transmission purposes. Essential to the support of group communication is the adoption of three-state switches and tap-couplers in the node architecture.

A slowly tunable λ -drop is used to separate the wavelength channel carrying the control slot from the wavelength channels carrying the payload slots. The control slot is converted to electronic domain and processed by the header processor. To account for the time to process a control slot, payload slots are delayed in fiber loops.

¹ For the sake of simplicity we assume N to be an integer multiple of W .

sense in the basic network architecture described in Sect. 2, MANs are usually laid out as counter-rotating ring topologies or interconnected ring topologies. In these topologies², TE is extremely important.

Specifically, the protocol follows the Multi-Protocol Label Switching (MPLS) architecture [4], which is under development within the Internet Engineering Task Force (IETF). Slots follow a previously established label-switched path (LSP). Each node along a LSP maintains a cross-connect table known as label information table (LIT). Each LIT entry identifies an input triple of the form <fiber link, wavelength, label> to an output triple of the same form. Each control slot is assigned a label at an ingress node. At each subsequent node, the label is used as an index to a LIT to determine whether the corresponding payload slot should be either received, forwarded or discarded.

Since control slots are processed at every node along a LSP, the layout of the control slot is very important to minimizing protocol latency. Fig. 4 depicts the control slot layout.

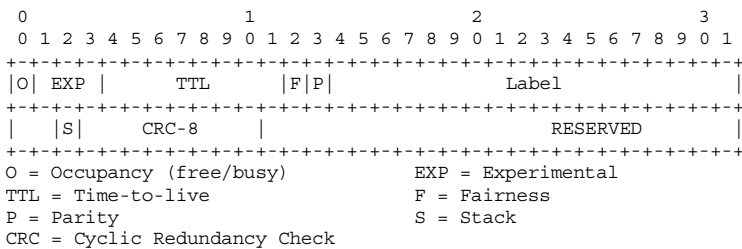


Fig. 4. Control slot layout

The MAC protocol provides a transport service. The payload slot layout is an opaque structure in which the MAC protocol has no interest whatsoever. The protocol simply transports a frame received from a source node's high-level data link control (HDLC) sub-layer to a destination node's HDLC sub-layer. This transport service is unreliable. Error probabilities in all-optical networks are so low that reliability can be better achieved at end nodes' HDLC sub-layers. Trying to achieve it in a hop-by-hop basis just adds death weight (i.e., useless overhead) to the protocol.

To achieve high throughputs and low delays, unicast slots are released at destination nodes and can be reused by the destination node itself or by any downstream node. This implies that transmissions on a given channel can occur simultaneously (as long as they take place on distinct links). The term spatial reuse is also used to characterize networks that employ destination release.

Multicast slots can be released either at the last destination or at the source. In the former, a multicast slot is forwarded over a point-to-multipoint LSPs. This is more suitable for networks running dense mode multicast routing. In such networks, receivers are densely distributed geographically and, therefore, releasing multicast slots at last destination nodes may result in considerable performance improvements. Salvador et al., in [5], propose a protocol to construct point-to-multipoint LSPs in all-optical WDM networks running dense mode multicast routing. In the latter, a

² We do not consider these topologies in this paper.

multicast slot is simply broadcast. This approach is simpler and more suitable for networks running sparse mode multicast routing. In such networks receivers are sparsely distributed geographically and, therefore, improvements that can be achieved in terms of performance are not worth the cost of complexity that is required to enable last destination release.

Destination release may introduce access unfairness under certain traffic patterns. A node that is immediately downstream to a node that is the destination of most of the traffic in the network may monopolize the channel and prevent other nodes from transmitting. To achieve access fairness, the transmission quota-based mechanism proposed in the MetaRing architecture [6] is adopted. Under such a mechanism, transmission only takes place if the three following conditions are met: i) there is at least one packet waiting for transmission; ii) an empty slot arrived and iii) the node still has some transmission quota left. Otherwise, the node refrains from transmitting and forwards the slot to the next node.

The fairness mechanism works as follows. Upon two visits of the so-called SAT signal, a node is allowed to transmit up to $l + k \geq 0$ data units, where l and k are multiples of the slot size. If upon visit of the SAT signal (i.e., the F bit of the incoming control slot is set to 1) a node has transmitted at least l data units, the node's quota is renewed and the SAT signal is immediately forwarded to the next node. Otherwise, the node holds the SAT signal (i.e., sets the F bit of the outgoing control slot to 0) until it has transmitted at least l data units. The node's quota is then renewed and the SAT signal is forwarded to the next node (i.e., the F bit of the outgoing control slot is set to 1).

Fairness is enforced on a node basis. However, there is one SAT signal regulating transmission on each channel. This is to prevent nodes transmitting on highly loaded channels from affecting nodes transmitting on lightly loaded channels. If only a single signal is used in the network, nodes transmitting on highly loaded channels will hold the SAT signal and nodes transmitting on lightly loaded channels will be prevented from transmitting while the SAT signal does not arrive.

Each node maintains one queue per LSP that the node can transmit over (i.e., each node maintains at least $N-1$ queues, where N is the number of nodes in the network). In the Internet, packet sizes are variable and mostly small [7] while the slots are of equally fixed size. Thus, with a single queue it may not be possible to achieve reasonably good slot utilization due to head-of-the-line (HOL) blocking.

Queues store frames rather than packets. Frames are formed in advance because performing framing on the fly may constitute a bottleneck at certain bit rates. A frame may contain one or more packets depending on the slot size and the packet sizes. Although concatenating packets at the source and separating them at the destination are expensive operations in terms of processing, we believe that this is acceptable considering that slots are of fixed size while packets are of variable size.

One could argue that because packets in the Internet are mostly small [7], small slots could be used so that concatenation and separation (CAS) operations would not be required. The price to pay for this simplicity, however, is higher control slot forwarding rates. This may constrain the network in terms of scalability [8].

A frame also contains a node's 48-bit MAC address. The source node's address can be used by the destination node's HDLC to inform that certain frames or even packets are missing.

The protocol works as follows. Upon arrival of a slot, a node first verifies if P is correct. If not, the corresponding payload slot is discarded and marked as free. If the

node can transmit on the payload slot's wavelength then it attempts transmission. If no error is detected, the node proceeds by checking O to find out whether the slot is free or busy.

If the slot is free and the node has sufficient transmission quota left, the MAC layer informs HDLC about the arrival of a free slot along with the slot's fiber link and wavelength. Based on the slot's fiber link and wavelength, HDLC moves the packet scheduler to the appropriate queue and selects the frame at the head of the queue. HDLC then returns the frame together with label that describes the LSP over which the frame must be sent.

The MAC protocol updates the control slot's label field with the received label. The other fields are updated accordingly as well. The control slot is forwarded to the next node. Transmission of the selected frame is delayed sufficiently to keep payload slots synchronized in parallel³. This assures that misalignment of slots due to dispersion is corrected at least once every ring revolution.

At each subsequent node, the slot's label is matched to a LIT. If no match is found, the corresponding payload slot is discarded and marked as free. If the node can transmit on the payload slot's wavelength then it attempts transmission. If a match is found that determines that the slot must be forwarded, the slot's label is swapped with the matched entry's outgoing label and the slot is forwarded over the outgoing interface and the outgoing wavelength. If a match is found that determines that the node is a destination of a multicast session, the payload slot's content is sent up to HDLC. The slot is forwarded to the next node according to the matched entry. If a match is found that determines that the node is the destination (in case of unicast) or the last destination of that slot (in case of multicast), the payload slot's content is sent up to HDLC and the slot is marked as free. If the node can transmit on the slot's wavelength, it attempts transmission. Otherwise, it forwards the empty slot to the next node.

4 Performance Results

We now present some performance results that were obtained via simulation activities. The simulations considered a 50km long network with $N = 16$ nodes, each equally spaced from one another, and $W = \{2, 4, 8, 16\}$ wavelengths. Slots are 552-byte long and packets are assumed to fit exactly in the slots. Transmission quota $Q(l, k)$, where $l = 100$ and $k = 200$, is chosen. Each node generates the exact same amount of unicast traffic to each other node. Packet arrival is Poisson and is such that every queue in each node has always at least one packet ready for transmission. Node 0 also generates multicast traffic at the same rate to nodes 1, 3, 5, 7, 9, 11.

³ Advanced signal dispersion is assumed.

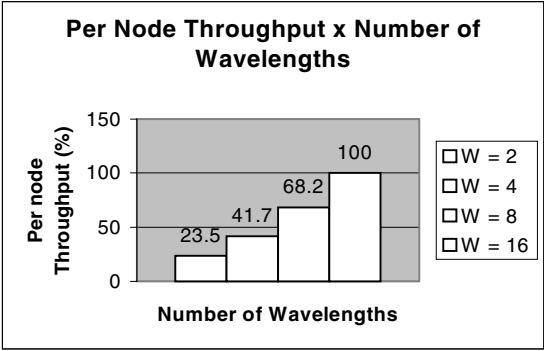


Fig. 5. Per node average Throughput

Fig. 5 plots per node throughput. Per node throughput is considered as the number of successfully transmitted slots divided by the total number of processed slots by a node. Fig. 5 shows that nodes achieve maximum throughput when $N \leq W$. Two are the reasons for this: i) there is no access contention since each node is assigned an exclusive transmission channel; and ii) $l \geq$ the bandwidth * latency product, which guarantees that a node will never be prevented from transmitting upon arrival of an empty slot. As the number of nodes sharing a given channel increases, the throughput of each of these nodes gets worse.

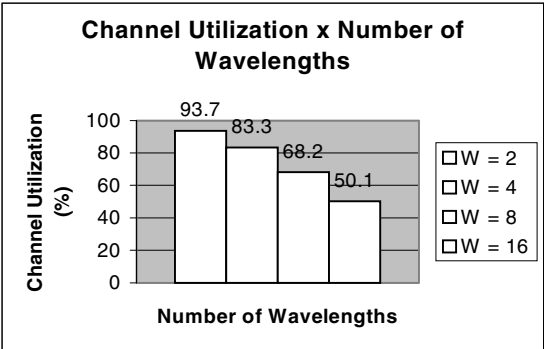


Fig. 6. Average Channel Utilization

Fig. 6 plots channel utilization. Channel utilization is considered as the number of links traversed by non-empty slots divided by the total number of links traversed by both non-empty slots and empty slots (plus inter-slot gap). As expected, the utilization of a channel improves as the number of nodes transmitting on that channel increases. The explanation for this statement is that under uniform and symmetric traffic conditions the average number of hops traversed by a (non-busy) slot from a source to a destination is $N / 2$. Thus, once a slot is released it has to traverse H hops in the average before being reused, where H is given by N divided by the number of nodes

transmitting on that channel times 2. This explains why, for instance, average wavelength utilization approximates 50% when $N = W$.

Note that this figures change in the presence of multicast traffic as there might be more than one destination. In this case, average channel utilization will depend on the distance in number of hops from a source to the last destination. The bigger the distance the higher the channel utilization is.

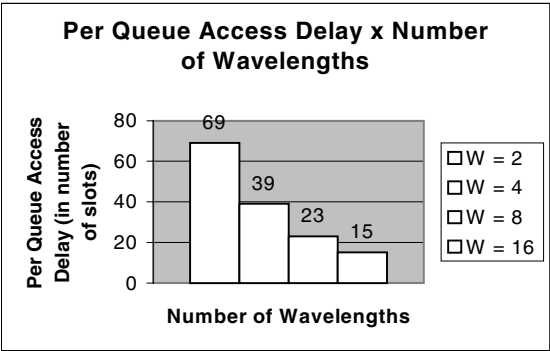


Fig. 7. Per Queue Average Access Delay

Fig. 7 plots per queue average access delay. Again as expected, per queue average access delay gets higher as: i) the number of possible destinations increases; and ii) the number of nodes transmitting on a given channel increases. When $N = W$, per queue average access delay is given by the number of possible destinations. This is why, for instance, when $N = W = 16$ and there are no multicast receivers, per queue average access delay equals 15 (a node does not transmit to itself and, therefore, the number of possible destinations or number of destination queues equals $N-1$).

Per queue average access delay gets higher as the number of nodes transmitting on a given channel increases. This is because access contention increases. Consequently, it takes longer for the packet scheduler to complete its cycle (i.e., to return to a given queue).

5 Concluding Remarks

This paper focused on a WDM MAN architecture that has been designed to support packet switching in the optical domain. Unlike others, this architecture supports not only point-to-point communication, but also point-to-multipoint communication. This is an important feature in the support of next generation services.

The network strives mainly for simplicity. Simplicity is fundamental in our network (even more than in conventional networks) because it leads to low processing delays. Fiber loops are directly proportional to processing delays of control slots. Thus, simplicity leads not only to scalability in terms of bit rate, but also to low cost due to the need for shorter fiber loops.

Simulation activities showed that the network provides high performance. Certainly, channel utilization is not the strength of the architecture, specially when

$W=N$. However, MANs may contain up to a couple of hundred nodes while the number of wavelengths is likely to be smaller. Furthermore, under certain traffic patterns (e.g., multicast), channel utilization improves. Finally, bandwidth is expected to be abundant and cheap and, therefore, channel utilization at the levels showed in this paper may not be an issue.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and technology programme of the Ministry of Economic Affairs of The Netherlands. The authors are also grateful to Kishore Kumar Sathyanandam for his cooperation to this work.

References

1. Marsan M. A., Bianco A., Leonardi E., Meo M., Neri F.: MAC Protocols and Fairness Control in WDM Multi-Rings with Tunable Transmitters and Fixed Receivers. *IEEE/OSA Journal of Lightwave Technologies*, Vol. 14, No. 6, June 1996.
2. Chlamtac I., Elek V., Fumagalli A., Szabo C.: Scalable WDM Access Network Architecture Based on Photonic Slot Routing. *IEEE/ACM Transactions on Networking*, Vol. 7, No. 1, February 1997.
3. Dey D., Koonen A.M.J., Salvador M. R.: Network Architecture of a Packet-switched WDM LAN/MAN. In *Procs. of IEEE/LEOS Symposium, Benelux chapter, Delft, The Netherlands*, October 2000.
4. Rosen E. C., Viswanathan A., Callon R.: Multiprotocol Label Switching Architecture. *IETF RFC 3031*, January 2001.
5. Salvador M. R., Heemstra S. de G., Dey D.: Supporting IP Dense Mode Multicast Routing Protocols in WDM All-Optical Networks. In *Procs. of SPIE/IEEE/ACM Optical Communications and Networking Conference, Dallas, Texas, US*, October 2000.
6. Ofek Y.: Overview of the MetaRing Architecture. *Computer Networks and ISDN Systems*, 26, pp. 817-829, 1994.
7. Thompson K., Miller G. J., Wilder R.: Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network Magazine*, November 1997.
8. Karn P. et al.: Advice for Internet Subnetwork Designers. *IETF Internet Draft <draft-ietf-pilc-link-design-05.txt>*, February 2001.

Performance Consideration for Building the Next Generation Multi-service Optical Communications Platforms

Siamak Dastangoo

OnexTM Communications Corporation, 34 Crosby Dr.,
Bedford, Massachusetts 01730, USA
(781)276-4137, sia@onexco.com

Abstract. The network service providers require cost-effective multi-service platforms that can meet their customer's diverse, dynamic, and demanding application requirements. Multi-service platforms require flexible, configurable, versatile, scalable, and multi-purpose VLSI solutions. The emerging ASIC solutions, for these applications, are appropriately termed Network Processors (NPs) or "systems on a chip." Many of the emerging NPs are limited to only processing cell/packet-based traffic with functionalities distributed over several chip components. Onex's intelligent TDM, ATM, Packet (iTAPTM) system can support TDM-based and cell/packet-based traffic on only two chip components. In this article, we provide a short background on the network processors followed by an overview of iTAPTM's system architecture. The iTAPTM's distributed dynamic bandwidth allocation mechanism will be described. A simulation model of some of the algorithms implemented in the iTAPTM is also provided. Finally, we will state the concluding remarks and elaborate on the issues that require further investigation.

1 Introduction

The explosive growth of the Internet traffic, the proliferation of the optical communications technologies, and the emergence of packet switching technologies (e.g., IP, ATM, and MPLS) have created new opportunities and demand for integrated, scalable, and easily configurable Network Processors (NPs) solutions enabling the next generation optical network platforms to provide integrated services. The legacy metro and access optical communications platforms are designed relying on custom-made ASICs (Application Specific Integrated Circuits) which are limited to a specific set of protocols or applications. For example, they typically only support ATM, or IP switching applications. Even, the emerging multi-service platforms have to rely on multiple native-mode processing elements and switching fabrics in order to support IP, ATM, and TDM services. These approaches lead to "fork-lift" upgrade scenarios which are not desired by the service providers. The service providers (e.g., CLECs and ILECs) require solutions that are highly scalable, configurable, extensible, and cost-effective.

The current philosophy towards the design and development of the network processors (i.e., system on a chip) is to design unified ASICs that can integrate all the

functions of the packet switching and circuit switching capabilities into as few chips as possible. Some of these functions include framing, mapping, traffic shaping, classification, routing, prioritization, queueing, and scheduling algorithms. The Onex iTAP™ system consists of two components, the service processor (SP) and the switch element (SE), respectively, where most of the traffic processing functions listed above have been implemented on the SP element. In this article, we provide some background on the requirements for Network Processors. Next, we discuss algorithmic and architectural issues specific to the Onex's iTAP™ system. Some performance studies of the iTAP™ chipset architecture are also reported.

2 Background

In the past decade, the research community has conducted numerous studies and achieved significant results in the formulation of sophisticated algorithms executing on-the-fly forwarding, classification, and scheduling techniques capable of enforcing quality of service (QoS) while maintaining high scalability. Switches that can scale to large capacities have a highly distributed architecture. Most of the functionalities, in these systems, are located in the ingress and egress port cards. Ease of implementation and cost-effectiveness of the distributed architectures have led to a series of highly-flexible and highly-scalable multi-stage, multimodule switch sets. The ATLANTA architecture, [1], with its memory/space/memory (MSM) arrangement brought the first commercial prototype of this type of architecture. The ATLANTA chip set was designed with the objective in mind that the dominant protocol of choice was ATM. The paradigm shift to IP and MPLS as well as the continuous need for the TDM-based services have created a demand for unified ASICs that can support all types of services.

Simultaneous support of TDM-based and packet-based traffic on the same Network Processor can be a complex task since the TDM-based traffic can not be queued and requires uninterrupted service through the switch fabric. The packet-based traffic can be queued and processed by various forwarding functions, however, a mechanism must be in place to adapt to changing bandwidth requirements without disturbing the TDM-based traffic. The packet arbitration mechanism developed for the iTAP™ system is designed for this purpose.

The packet arbitration mechanism and the scheduling schemes in the line cards (ingress and egress) assure that packet traffic receive adequate resources while meeting the required QoS guarantees. The purpose of the study in this paper is to investigate performance issues related to these schemes and the interaction between the arbitration and scheduling mechanisms. In the following, we will briefly review some of the packet processing functions performed in the iTAP system. We also highlight several packet scheduling techniques used here. A detailed description of the iTAP™ packet arbitration mechanism will be provided in section 4.

2.1 Packet Processing Functions

In this section, we use packet processing as a generic term that describes different actions applied to a packet once it enters and exits a Network Processor.

2.1.1 Traffic Forwarding

Traffic forwarding function performs layer-3 IP packet forwarding. By forwarding, it is implied the process by which a routing database is consulted to determine which output link, a single packet should be forwarded on. This process of database consultation must be extremely fast to keep up with the rate of arrival of packets.

2.1.2 Traffic Classification

IP enhanced services such as differentiated services (DiffServ), fine grain quality of services, virtual private network (VPN), distributed firewalls may require different treatment as they enter the IP network. These services need packet classification, which is the process of determining which flow a packet belongs to, based on one or more fields in the packet header. In the packet classification process, header fields that can be used consist of the destination and sources IP addresses, the protocol type, and the source and destination port numbers. By specifying valid ranges for any of the header fields, different rules for classification can be established. The iTAPTM service processor using intelligent classification algorithms can easily support this feature at OC48 link rates, [6] and [7].

2.1.3 Traffic Policing and Congestion Management

Traffic policing is the process by which incoming traffic is examined to determine whether or not it conforms to the negotiated traffic contract. During this process those packets that have violated the traffic contract must be appropriately identified, marked or dropped. Congestion management is the process in which the nonconforming traffic is dropped. The dropping process can be achieved in a way that allows the user applications to adapt their transmission rates to network congestion conditions. For the congestion management function, the iTAPTM system implements variations of the weighted random early detection (WRED), [8], scheme to alleviate possible congestions.

2.1.4 Packet Scheduling

Packet scheduling is the process of time-stamping packets according to their priority for departure on the output link. There are several packet schedulers, these are described in the following sections.

2.1.4.1 Weighted Round Robin

Weighted round robin (WRR) is a simple scheduling scheme with very low implementation complexity that enforces accurate bandwidth guarantees at the expense of excessive delay bounds. WRR is used for scheduling request elements for transmission of packet data units (PDUs) through the switch fabric and the egress service processor. More discussion on this scheduler will follow later.

2.1.4.2 Generalized Processor Sharing

A generalized service processor (GPS), [2], is a service mechanism with the following attributes:

- Minimum bandwidth guarantees,
- Deterministic end-to-end delay bounds,
- Fairness in the distribution of service among different flows

A formal characterization of generalized processor sharing defines K positive real numbers, $\phi_1, \phi_2, \dots, \phi_K$ where the server operates at a fixed rate r and is work conserving. Define $\Psi_i(\tau, t)$ to be the amount of session i traffic served in an interval $[\tau, t]$. A GPS server is defined as one for which

$$\frac{\Psi_i(\tau, t)}{\Psi_j(\tau, t)} \geq \frac{\phi_i}{\phi_j} = 1, 2, \dots, K$$

for any session i that is backlogged throughout the interval $[\tau, t]$. Any session i is guaranteed a rate of

$$\delta_i = \frac{\phi_i}{\sum_j \phi_j}$$

Since GPS is an idealized discipline that does not transmit packets as entities, a packet-by-packet approximation to the transmission scheme is needed. A weighted fair queueing (WFQ) scheduler can be used to emulate the bit-by-bit GPS. Several variations of weighted fair queueing have been investigated and implemented over the past decade [3], [4], and [5]. A computationally efficient variation of WFQ which achieves fairness regardless of variation in server capacity is called start-time fair queueing (SFQ), [3].

3 System Architecture

Figure 1, illustrates a high-level architecture of the Onex iTAP™ system. This system consists of two components, the service processor (SP) and the switch element (SE), respectively. The service processor is designed to SONET OC-12 and OC-48 interfaces as well as UTOPIA level 3 for ATM and POS and can be extended to higher (e.g., 10 G.) The switch element is capable of supporting up to 12 ingress and 12 egress service processors which can aggregate to 30 Gbps of throughput and up to 10 Tbps if a multi-stage configuration is used.

The service processor, once properly configured, can simultaneously process packet-based and TDM-based traffic. The provisioned TDM traffic flows through the switch element without being disturbed by the statistically multiplexed data traffic. The packet-based traffic (e.g., IP, MPLS, ATM), on the other hand, must go through several stages of packet processing. Upon the arrival of data traffic in the ingress service processor, the packet/cell headers are extracted for further processing while the payload is queued up to be scheduled through the switch element and the egress service processor. The first processing phase for packet/cell is layer 3 IP address lookup/ layer 2 label/VCI/VPI table lookup. Immediately after that, the classification process determines which flow a packet belongs to based on one or more fields in the packet header.

The policing and congestion management processes take place immediately where IP packets and ATM cells are usage parameter controlled for conformance checking.

If for any reason traffic contracts are determined to be violated, the policer must enforce appropriate actions which include tagging of the violating traffic. The responsibility of the congestion management process is to enforce the traffic contract rules under which the violating PDUs are discarded using WRED scheme.

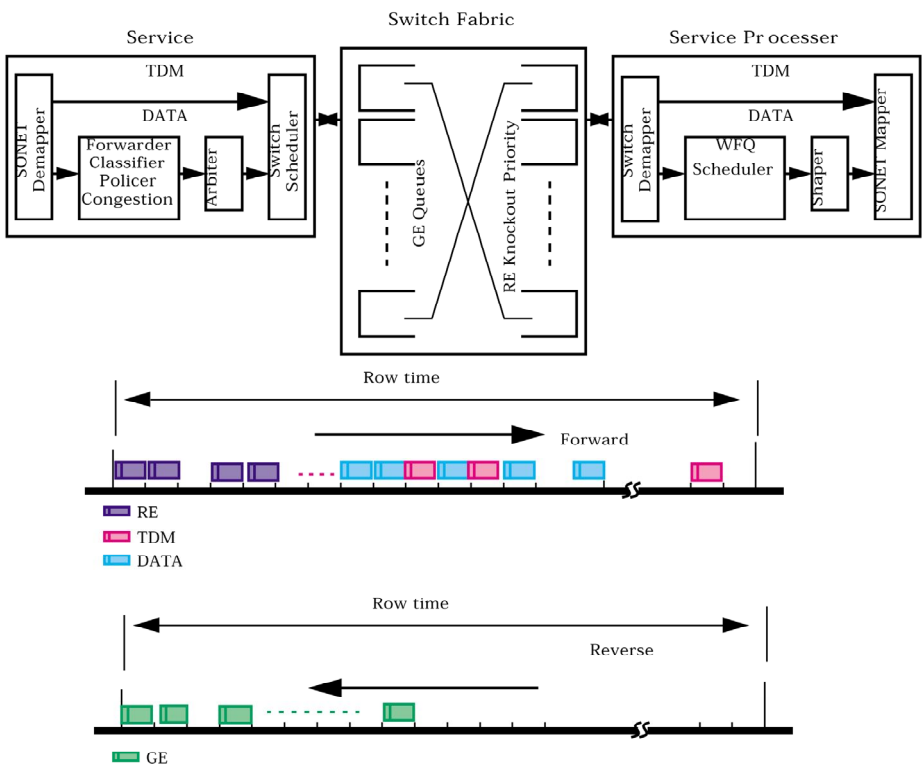


Fig. 1. High level architecture of iTAP™ network processors and switch fabric.

All of the received traffic is destined for the Switch fabric and is mapped into Onex's proprietary row format, see Figure 2. This row consists of 1700 slots of length 36 bits each. Each slot carries 4 bytes of data and 4 bits of control information. TDM-based data is allocated dedicated bandwidth through the switch fabric and the Onex proprietary row format is designed to optimally support TDM traffic down to the VT-1.5 and VC-12 level. Incoming TDM traffic is never buffered, it is routed directly to pre-allocated and pre-configured slots in the outgoing rows. TDM data may be switched through the SE with a finest granularity of 1 slot per row. This means that a TDM stream may be switched from any slot on any input link to any slot on any output link. ATM cells, IP Packets, MPLS packets, and PPP frames are not allocated dedicated bandwidth through the switch fabric. The bandwidth for data services must be arbitrated through the switch and the egress Service Processor. The row is designed to support a super-slot or data-slot which is an aggregation of 16 single slots. The switch row is serialized and distributed across high speed serial ports for transmission to the switch.

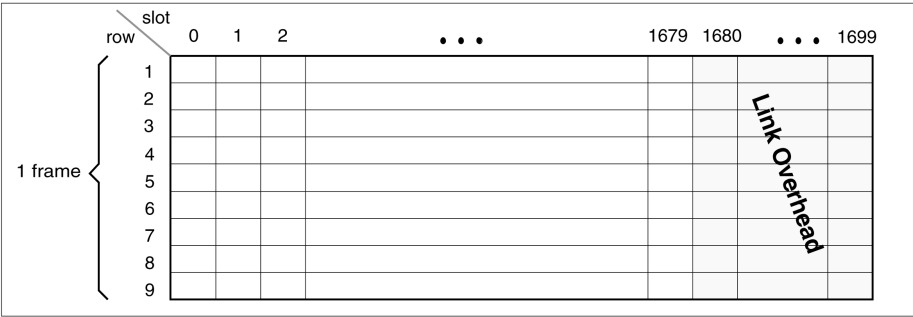


Fig. 2. Frame structure.

In the Transmit direction, the Service Processor responds to arbitration requests from a Switch chip. The arbitration grant from the Service Processor is based on the available buffer space in the traffic memory. TDM traffic and granted data traffic are received through the high speed switch interface. TDM traffic is mapped directly into outgoing SPEs. ATM Cells, and IP packets, MPLS packets, PPP frames are all buffered externally where they wait to be scheduled out a transmit SONET or UTOPIA interface.

4 Arbitration Mechanism

4.1 Distributed Dynamic Bandwidth Allocation

Each service processor operates without any fore-knowledge of what the other service processors are doing. As a result, when they go to send their PDUs, they need to know two things, see Figure 3:

- Does the egress service processor have room in its queues for this PDU?
- Is there bandwidth in the chosen path to get the data from one end of the switch fabric to the other without packet loss?

The arbitration mechanism will check both of these criteria and send back a grant message to the requesting service processor on a PDU by PDU basis. When a service processor has received a grant it knows for certain that the data will make it to the egress service processor (except during system failure).

Each row time, the service processors will make a request for each PDU to be sent in the next row. There can be up to 96 request elements per row time, one element for each possible data PDU. These request elements will be multiplexed in with the data stream. The requests stream through the switch. Contention resolution is achieved via a priority based knockout mechanism. This is important since the 12 inputs could all converge on a single output, the outgoing link will not be able to handle the traffic presented to it. A small buffer pool exists in each output link to hold some of the requests when multiple requests come into the switch chip which are destined for the same output link. At the far end of the switch fabric, the service processor will make a decision to grant or deny a request based on the depth of its QoS queues. The service processor will then source a grant message which also travels through the switch

fabric, but in an *out-of-band* overlay network which goes in the opposite direction of the switch fabric. The grants will be written without regard to priority into a first-in-first-out (FIFO) queue and read in order of arrival time.

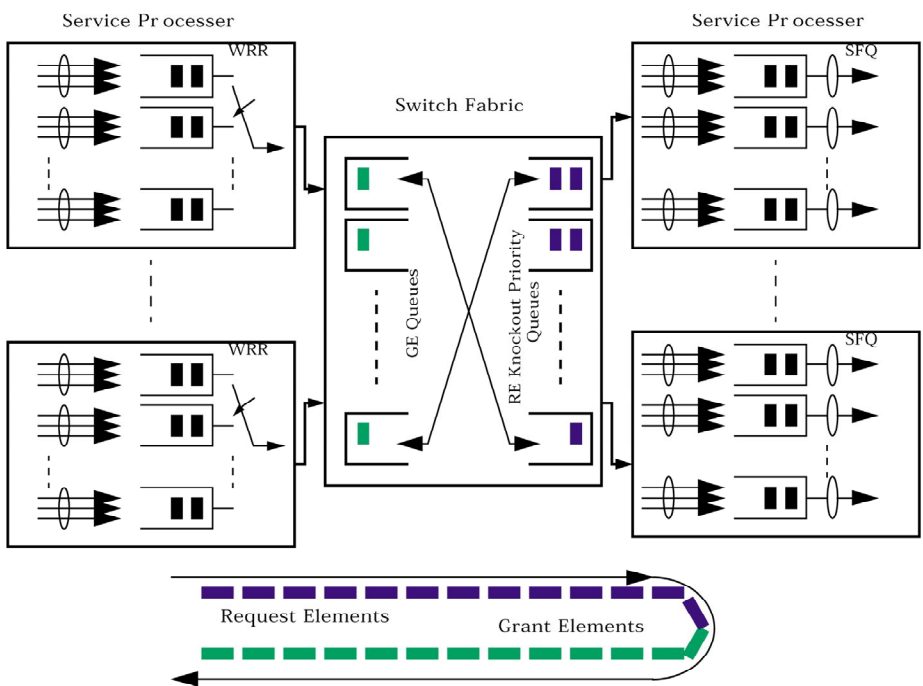


Fig. 3. Queuing model of the iTAP system service processor and switch element.

The arbitration mechanism will work as follows:

- At the start of a row time the input service processor will begin sending its requests. A request element is a request for a single group's worth of bandwidth in the switch fabric destined for a particular service processor,
- The first stage in the switch fabric will look at each request from all 12 input links as well as the multicast and control message controller. The requests traverse the switch fabric by using a self routing tag which indicate the hop-by-hop output ports used at each stage of the switch fabric. At this time, the Stage 1 hop-by-hop field will be replaced with the input port number that the request entered on,
- The requests for each output link will be stored in a buffer pool. As long as buffers are free, requests will be stored. As soon as there are no free buffers, lower priority requests will be overwritten with higher priority ones. The request buffers will be able to support 12 input links all converging on a single output, meaning that 12 request elements can be written to the buffer pool every request time. Requests are evicted from the buffer pool based on priority and age. The youngest lowest priority requests will be dropped, and the highest priority oldest requests will be kept,

- The request buffers will be then read. After a request is read the request element deleted from the buffer, making room for another request element. Any requests still in the buffers will be dropped,
- This process continues all the way to the end of the switch fabric and into the service processor. The service processor will make a decision to accept or reject the request based on the QOS field. Then, it will source a grant message. The grant message uses the modified self routing tag of the request element to traverse the switch fabric backwards using an overlay network,

In section 5, we describe a simulation model of the scheduling and arbitration mechanisms used in the iTAPTM system.

5 Simulation Model of iTAP System

In this section, we briefly describe the model of the iTAPTM system constructed on the Opnet simulation platform. The primary purpose for this study is to observe the interactions among four major components of the iTAPTM system the packet scheduling on the ingress service processor, packet arbitration mechanism, the knockout priority queue mechanism in the switch element, and the packet scheduling on the egress service processor. The specific algorithms used for these disciplines are described below.

The service processors include:

- Weighted round robin (WRR) scheduler,[1], to schedule the packet-based traffic in the ingress Service Processor,
- Weighted fair queueing (WFQ) scheduler, [2], to schedule packet-based traffic on the egress ports,
- Egress port buffer management,
- Packet arbitration mechanism.

The switch element consists of:

- A knockout priority queueing mechanism to prevent possible congestions when multiple ingress ports converge onto the same egress port, [4].

On the left side of Figure 4, a detailed description of the WRR scheduling for the request elements and their possible location with reference to a the row format are depicted The knockout priority buffer for the request elements in the switch element is shown on the right side of Figure 4. A view of the Opnet model of the iTAPTM system is illustrated on the left side of Figure 5. In this study a single-stage switch fabric is shown with interfaces to 12 ingress and 12 egress service processors. The buffer management thresholds at the output queues in the egress service processor is illustrated on the right side of Figure 5.

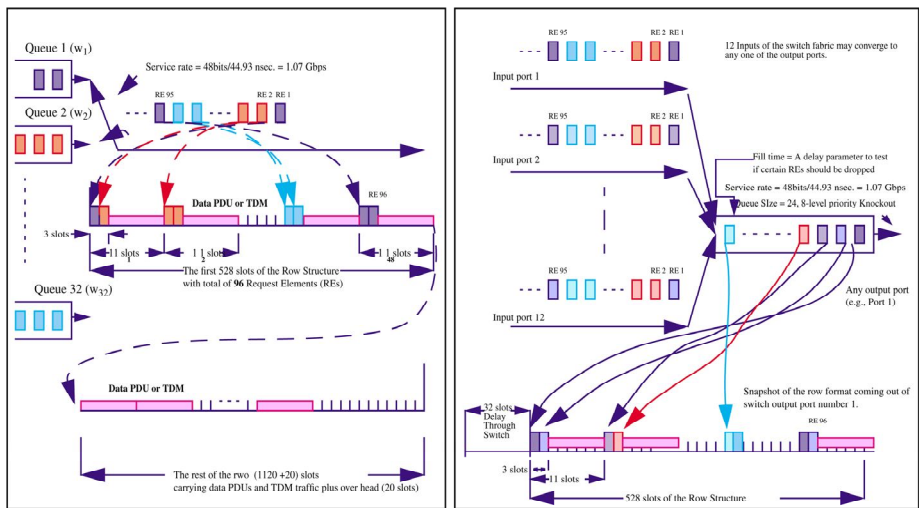


Fig. 4. Weighted round robin scheduling in the ingress service processor. knockout request element queue in the switch element.

5.1 Highly Overloaded System

In the following, we consider a scenario where the system is being observed where input generation rate is much larger than the service (drain) rate. In this experiment, three flows are configured to generate PDUs on average at 800 Mbps each according to a Poisson process. Each flow is assigned different QoS value where the QoS values determine the treatment each flow receives according to their predetermined weight. These flows are originated from three different ingress ports and are destined to the same egress port, see Table 1 for more details. The service rate at the output port is set to 800 Mbps. In this highly overloaded scenario, we observe the performance of the system when the system is subject to congestion. Initially, the flows are assigned the same weights (e.g., WRR weights) at the ingress side, the same switch knockout priority, and different weights (e.g., weights at the egress side) at the egress side. The next experiment, different switch knockout priorities are used to allow the higher priority flows to move through the switch. In the third experiment, we assign different WRR weights in the ingress service processor. We expect to observe performance improvement for higher priority flows. Figure 6 illustrates time average of flow throughput for three flows described above.

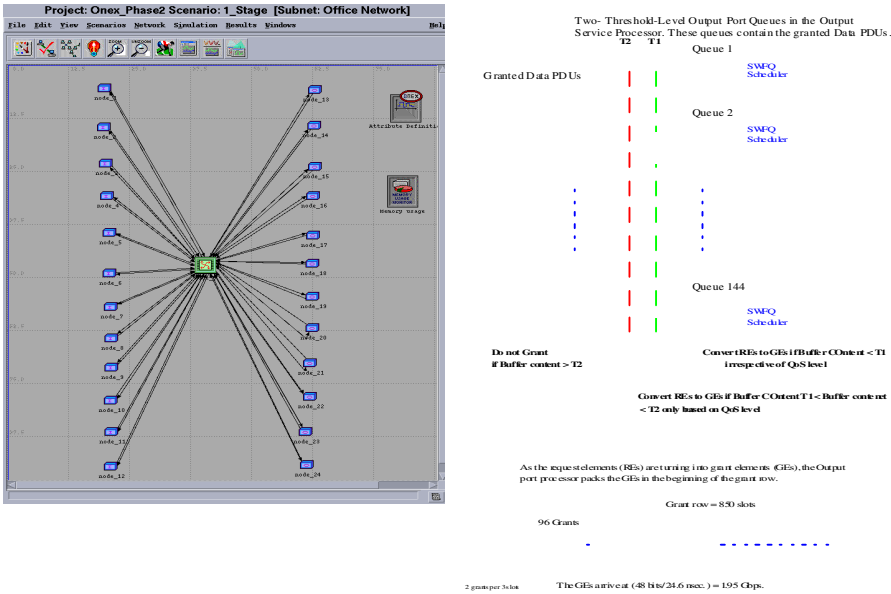


Fig. 5. Network view of the iTAP™ system model in Opnet simulation platform.

Table 1. Flow configuration for the simulation experiment.

Number of Flows	Generation Rate/Flow	Drain Rate	Queue Thresholds	QoS For Low Thresh
3	800 Mbps	800 Mbps	(200,500) PDUs	>1

As shown in Figure 6(a), three curves represent the throughput behavior for three flows. The two high QoS flows number 0 and 1 receive larger share of resources (e.g., bandwidth at the egress port) as compared to flow 2. The abrupt reduction of the throughput for flow number 2 is due to the buffer management threshold used in the output queue. In Figure 6 (b), as different knockout priorities are used in the switch element, we can readily observe the its effect on flow number 0 improving its performance since the request elements for flow number 1 and 2 are knocked out more often by the switch element. Finally, when different WRR weights are assigned to the test flows, the flow (e.g., flow number 0) with the highest weight can receive better performance, see Figure 6(c). It is also interesting to observe that the aggregate throughput of the three test flows sums to 800 Mbps which is equal to the egress side drain rate.

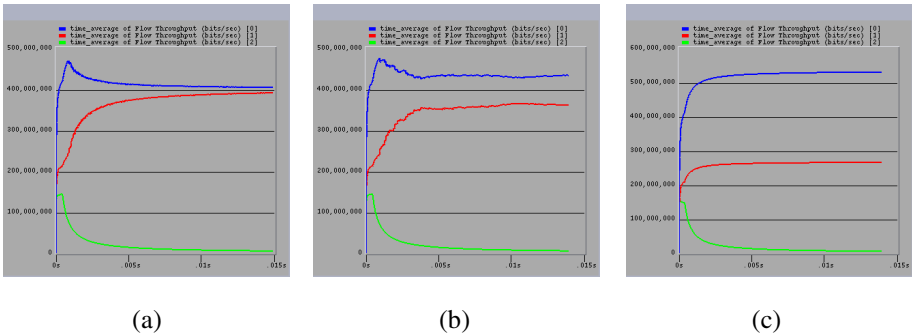


Fig. 6. Time average of flow throughput for three flows with SFQ, knockout priorities and WRR. a) SWFQ only, b) Knockout priority, and SWFQ, c) WRR, Knockout priority and SWFQ.

6 Conclusions

In this article, we presented the architecture for the Onex’s iTAP™ Network Processor. These processors can provide packet-based and TDM-based functions suitable for multi-services platforms. An arbitration algorithm used for packet-based traffic is stated. An event-driven simulation model of the scheduling and arbitration algorithms were developed on the Opnet environment. The preliminary results indicate that the mechanisms built to handle data services in conjunction with TDM services achieve desirable performance in terms of user throughput and end-to-end delay bounds.

References

1. Chiussi, F. M., et al, “Low-cost scalable switching solutions for broadband networking: The ATLANTA architecture and chipset,” *IEEE Commun. Mag.*, vol. 35, pp. 44-53, Dec. 1997.
2. Parekh, A. K. and Gallager, R. G., “A generalized processor sharing approach to flow control in integrated services networks: The single-node case,” *IEEE/ACM Trans. Networking*, vol. 1, pp. 344-357, June 1993.
3. Goyal, P., Vin, H. M., and Cheng, H., “Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks,” *IEEE/ACM Trans. Networking*, vol. 3, pp. 690-704, October 1997.
4. Walrand, J. and Varaiya, P., *High-Performance Communications Networks*, Morgan Kaufmann, 1996.
5. Stephens, D. C., Bennett, C. R., Zhang, H., “Implementing scheduling algorithms in high-speed networks,” *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1145-1158, June 1999.

6. Feldman, A. and Muthukrishnan, S., "Trade-offs for Packet Classification," *IEEE INFOCOM*, 2000.
7. Lakshman, T. V. and Stiliadis, D., "High-speed policy-based packet forwarding using efficient multi-dimensional range matching," *ACM Computer Communication Review*, 28(4): 203-214, 1998. ACM SIGCOMM'98, Sept. 1998.
8. Floyd, S. and Jacobson, V., "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol.1, no.4, pp. 397-413, August 1993.

Traffic Management in Multi-service Optical Network

Halima Elbiaze and Tülin Atmaca

Institut National des Télécommunications

9 rue Charles Fourier

91011 Evry CEDEX - France

Phone : +33 1 60 76 47 42

Fax : +33 1 60 76 47 80

{Halima.Elbiaze, Tulin.Atmaca}@int-evry.fr

Abstract. The goal of this paper is the traffic management in the multi-service optical network context (ROM). We suggest centralising the traffic management policies at the interface between the client layers and the ROM network. The key idea is then to exploit, in an optimal way, the electronic memories in the electronic interfaces at the edge nodes of the optical network to control the incoming traffic.

We particularly study the impact of traffic shaping at the ROM periphery on the end-to-end performance in terms of loss and delay.

1 Preliminaries and Problem Relevance

With a world wide coverage giving access to a large range of data banks and a massive introduction of PCs in house capable to handle IP connections, it is foreseen that interactive applications are identified as a future important market. Well understood by the IP community, efforts are currently devoted to propose new techniques adapted to IP to offer a new Quality of Service (QoS). In addition, incumbent and emerging carriers have adopted a mix of Sonet/SDH, ATM, and IP. There is a clear agreement that the optimal solution delivers efficiently voice and data transport; is scalable, flexible, cost-effective, and reliable; while offering QoS. Optics has been identified as a technology capable to provide large capacity of transport in point-to-point transmission systems. Optical cross-connects have been studied for several years to propose an all-optical architecture capable to manage optical wavelength channels. However, the sporadic nature of the Internet traffic pushes equipment constructors to envisage all-optical packet switching networks to provide the flexibility required together with the capacity [1]. Thus, the objective of French ROM project¹, is to demonstrate the feasibility of a multi-service optical network where optics provide the underlying network architecture, compatible with a QoS requested by client layers at the

¹ Short for “Réseau Optique Multiservice”. ROM is partially supported by the “Réseau National de Recherche en Télécommunications” under the decision number 99S 0201-0204

periphery of the considered optical layer[2] . The lack of optical memories and the need to limit the use of electronic memories argues in favour of the physical resource naturally offered by optics: the wavelength dimension. In the context of broadband networks, with bandwidth up to or larger than 1Tbit/s, the interface between the electronic and optical sub-networks poses challenging issues. This interface shall conciliate and interconnect these two worlds. This paper focuses on end-to-end traffic engineering as a means to end-to-end traffic management between the client layer and the ROM network in the presence of different traffic types with different QoS requirements. We particularly study the impact of traffic shaping at the ROM periphery on the end-to-end performance in terms of loss and delay. The self-similar nature of Internet traffic has been demonstrated by several measurements and statistical studies [5]. It has a direct impact on network dimensioning; in particular buffer sizing is crucial. On one hand, the buffer size should be large enough to absorb the very long traffic bursts introduced by self-similar characteristics; on the other hand, the size should not be so large as to introduce unacceptable delays. In this paper, the proposed solution when designing the network is to increase the buffer size at the admission points in order to smooth out the peaks and valleys; and to dimension the optical links and memories so that the optical network can operate at high peak-to-average load. Thus, considering the intrinsic characteristics of the IP-based traffic, such as the asymmetric traffic flow, the traffic burstiness, and the self-similar or fractal nature of the traffic statistics, the Electro-optical interface has to include several traffic management techniques. These are mainly packets buffering with possible priority schemes, congestion control, shaping and policing.

In order to optimise the logical performance at the network level, and due to the lack of optical memories, the distributed memory is, in many aspects, the only solution. The key idea is then to exploit, in an optimal way, the electronic memories in the electronic interfaces at the edge nodes of the optical network to classify the incoming traffic, to wait for a fulfilling of payloads in order to optimise the transport, to reshape the traffic profile and to regulate the traffic in case of strong contention localised in the network.

2 Trends of Optical Multi-service Network Relevant to Our Study

2.1 Client Network

Subscribers ask for high QoS at low costs and offer a traffic mix consisting of data, video, and audio. This pushes network providers to provision an optical multi-service network at the core of electronic-based customer LANs (see Figure 1). At the electronic functional layer, traffic aggregation, packet routing, and traffic policing (access control at the interface level) are performed. At the internal, optical layer, the focus is on transmission and low-layer switching functions. The edge switches are located at the boundary between these two layers.

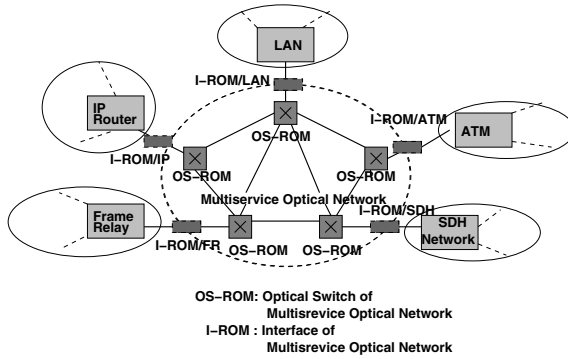


Fig. 1. The multi-service optical network structure

2.2 CoS/QoS and Optical Bandwidth Allocation

User applications have different characteristics and require different QoS levels. In this context the network should be able to offer a CoS/QoS to differentiate between the different application requirements. In this work, we consider three QoS levels:

- QoS level 1, for strictly real-time services, with very low packet loss rate ($< 10^{-9}$), strictly bounded delay and without loss of sequence in the optical nodes.
- QoS level 2, for less strict real-time and priority non-real time services, with low packet loss rate (e.g. $< 10^{-6}$), and without loss of sequence.
- QoS level 3, best effort for non-real time services, with a packet loss rate monitored by the client layer protocol (IP in this study), without control of delay and packet sequence.

The most efficient management of the bandwidth can be achieved by using long packets covering all the available wavelength channels [6]. This way, classical cross-connected optical transport network migrates towards a pure optical packet network.

2.3 Interfacing Issues

Very often, customers produce IP flows over various underlying infrastructures, such as ATM, FR, etc. At the optical interface, the interoperability issue can be simplified by considering the IP layer only and not the lower links. IP packets only are to be processed and IP QoS paradigm applies. The IP/optical interface brings challenging issues, mainly with respect to bringing together the electronic and optical technologies. The first issue concerns the bandwidth adaptation. Today, classical LAN solutions carry traffic in the range of 100 Mbit/s to 10 Gbit/s. On the other hand, on a single optical fibre, bandwidth is available under the form of separate wavelengths, each of which able to carry at least 2.5 Gbit/s

and up to 40 Gbit/s. The second issue is related to the link and the network layer functions. Classical network architectures rely on well-known traffic management techniques: packet buffering in the network node, with a possible priority service scheduling and congestion management. The all optical core is unable to offer the same traffic management capabilities, due to the limited optical buffer sizes.

2.4 Format of Optical Packet

We consider packets of fixed length, as shown in Figure 2, which opens the way to shaping techniques, able to dramatically reduce the burstiness level of a traffic profile. With a better traffic profile, the contention resolution in the optical nodes is relaxed and paves the way to the introduction of all-optical packet switching nodes while offering a high level of QoS. The structure of the optical packet is

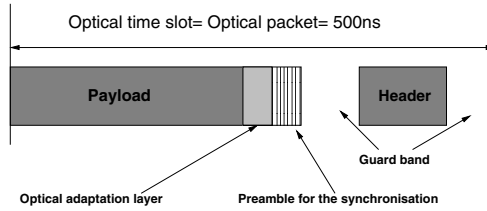


Fig. 2. Structure of the optical packet

composed of three main sub-blocs:

- Header - includes source address, destination address, priority, number of jumps, HEC, delineation, synchronisation,
- Payload - includes some bits as a preamble to identify the position of the payload and to ease the packet jitter extraction,
- Guard bands - inserted to help the switch read the relevant packet information while coping with length variations, thermal and chromatic dispersion effects etc.

3 Model of Study

3.1 Description

The end-to-end model is shown in Figure 3. The external part, at the customer side, consists of a set of traffic sources with different types of traffic. These are real-time voice and video conference, video-on-demand and classical data (WWW, FTP ...), corresponding to the three levels of QoS, respectively, and are to be differentiated according to those levels of priority. We investigate in this work the importance of traffic shaping to improve performance in the core of the optical network. We specifically target acceptable and feasible sizes of fibre delay

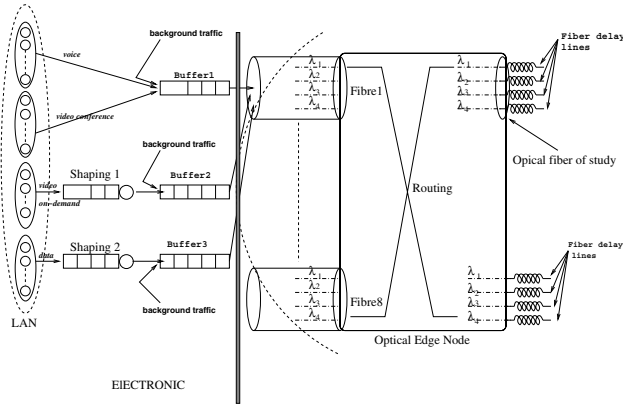


Fig. 3. Model of study

lines. As video-on-demand and classical data traffic have a self-similar nature, upon aggregation, the long-range dependence of this self-similar traffic tends to be even more apparent. This type of traffic when offered to an optical network which typically have little provision for buffering results in a bad performance at the optical level. The remedy consists in shaping the traffic at the interface, which is outside of the optical functional area, where memory comes cheap. Hence the novelty of the model we suggest lies in shaping of the incoming flows as they arrive from the customer and in decentralising the traffic management and conditioning at the periphery outside of the optical network. The latter is unable to perform those vital tasks but is solely capable of fast transmission. This done, incoming electronic packets are assembled in optical packets at the Electro-optical interface. Let us recall that in the ROM project, the edge node, where the interface lies, is modelled by a set of control mechanisms: i) conditioning and shaping of incoming traffic from the client layer, ii) differentiation, classification and buffering of the different flows into separate buffers according to their level of QoS, iii) priority scheduling mechanism, for instance using a head of line scheme, to take into account the different levels of priority, iv) filling of optical packets by incoming of incoming variable size IP packets which may be segmented if need be. The ROM edge node studied in this paper is a metropolitan optical node with 8x8 port and 4 λ 's per port, each wavelength carries 10 Gbit/s. The fibre delay lines associated to each output wavelengths are dimensioned as well as the electronic buffers implemented at the interface level.

3.2 System Parameters

The self-similar traffic parameters are illustrated by table 1. This values corresponds to mean throughput, standard deviation, Hurst parameter and time scales used for synthesising self-similar sources behaviour for video conference, video-on-demand and classical data. Voice sources are modelled by ON/OFF

process with 0.4s exponentially distributed ON duration, and 0.6s exponentially distributed OFF duration. Each voice source emits at 32Kbit/s. Table 2 shows packet sizes and allocated bandwidth for each CoS considered in our model.

Table 1. Self-similar parameters for each type of traffic

Type of traffic	video conference	video-on-demand	data
mean	181 Kbit/s	2 Mbit/s	200Mbit/s
standard deviation (σ)	35.10^3	10^5	10^7
H (Hurst)	0.51	0.8	0.9
time scale	3	5	6

Table 2. Bandwidth rate allocated to the 3 CoS considered

Classe of Service	CoS1	CoS2	CoS3
Packets size	160 octets	256 octets	1500 octets
allocated bandwidth	727 Mbit/s	5.2 Gbit/s	13 Gbit/s
Sources number	19452	2600	65
Percentage/total bandwidth	3.84 %	27.49 %	68.68 %

4 Results

Next, we have simulated the above-mentioned model. We investigate two scenarios, with and without shaping of video-on-demand and classical data traffic. We aim to study the dimensioning of the fibre delay lines subject to loss rate constraints and the trade off between shaping parameters, in terms of buffer size and shaping rate, and the sizes of the fibre delay lines necessary to meet the given level of performance.

The first issue is the filling of optical packets with incoming electronic ones is not always straightforward. Optical packets need to be filled with packets belonging to the same QoS level as the optical transmission cannot distinguish between payloads of the optical packets. In case of large incoming packets, we can simply segment them. However, if not enough packets are ready to be transmitted on the optical side, these results in under filled optical packets with in turn results in under utilised optical network.

We hence investigate the optical bandwidth utilisation versus the electronic packet size, taking into account the cost of packet segmentation at the destination level.

The second issue deals with the shaping of incoming traffic at the Electro-optical interface.

4.1 Impact of Shaping on Performance

Figure 4 shows the probability density function with respect to number of optical packets on one fibre delay line for the cases of shaping and no shaping. Both curves decrease as the number of optical packets increase showing higher density at the lightly loaded side of the fibre delay line. The no shaping curve however shows an extremely high density even at the 20 packet level which is beyond feasible cases for fibre delay lines. This shows that without shaping, our loss rate commitments cannot be met. In the case of shaping, the curve decreases more rapidly than in the no shaping case reaching a value of 10^{-8} at almost 7 optical packets at the fibre delay line. This shows that shaping only helps achieve the desired performance level that meets the QoS constraint

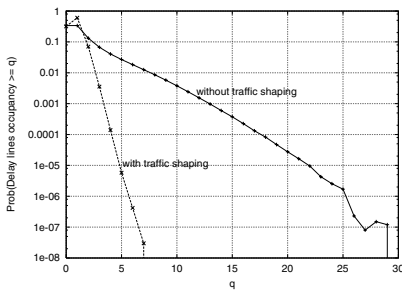


Fig. 4. Probability Density Function of one output fibre delay line

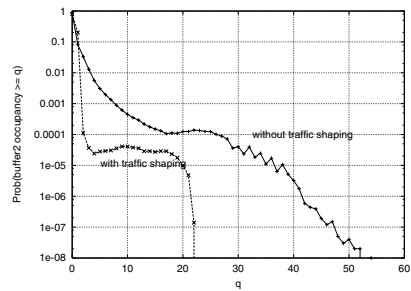


Fig. 5. Probability Density Function of buffer after shaper: buffer2

Moreover, shaping the data traffic maintains the end-to-end delay in the network while contributing largely to the improvement of the logical performance in terms of packet loss, as shown in [4]. Continuing with dimensioning, we next investigate the sizes of the buffer just after the shaper at the Electro-optical interface. Let us recall that those buffers are used to differentiate and classify flows according to their QoS levels. We again plot the probability density function with respect to number of electronic packets on each buffer. Figure 5 shows the probability density function with respect of the number of packets on buffer 2, assigned to video on demand traffic with medium QoS and thus priority. The no shaping curve shows high density at the 22 packet level, the point at which the loss rate will be not less than 10^{-4} , we need about 40 packets buffer space to reach our loss commitment. In the case of shaping, the curve decreases more rapidly than the no shaping case reaching the value 10^{-6} loss rate at 22 packets at buffer. This improvement of performance comes by implementing buffer shaping at the interface level. In our study, we have considered 2000 packets shaping buffer size for CoS2 traffic (referred by shaping 1 in figure 3) and 1000 packets shaping buffer size for CoS3 traffic (referred by shaping 2 in figure 3). The observed loss in shaping 1 (respectively shaping 2) is 10^{-5} (respectively 10^{-4}). We have next measured mean delays at the optical node and the end-to-end.

The obtained results are shown in table 3. The traffic shaping improve clearly delay at the optical level for different CoS.

Table 3. Mean delays

Mean delay	at the optical node		end-to-end	
	without shaping	with shaping	without shaping	with shaping
CoS1	800 ns	516 ns	1.310 ms	2.11 ms
CoS2	770 ns	513 ns	1.87 ms	2.84 ms
CoS3	1380 ns	569 ns	3.48 ms	7.9 ms

The confident level associated to the delay values is 95%. In the case of shaping, we have obtained between 0.4% and 0.8% for different CoS. But the no shaping case gives larger confident intervals (around 30%) because of high variation of self-similar traffic in different large time scale.

4.2 Optical Bandwidth Utilisation

In the above study, we have scheduled the electronic packets at the optical access according to HoL (Head of Line) policy. Every time slot, we fill the available optical packets with CoS1, CoS2 and CoS3 respecting the priority. In this section, we focus on the comparison between this resource allocation mechanism, which will be refered as **total flexibility** mechanism, and another one **partial flexibility**.

Partial flexibility mechanism consists on dedicating λ_1 only to CoS1 traffic. CoS2 et CoS3 will share λ_2 , λ_3 and λ_4 with respect of priority. The aim is to investigate the performance at the optical node level for both **partial flexibility** and **total flexibility**. One way to accomplish this task is to vary the CoS1 traffic percentage of the total traffic and compare the performance of these two mechanisms . We consider the same system model studied above assuming that all traffics are already shaped before acceding to the optical network and we vary the CoS1 traffic percentage from 6% to 66%. Figure 6 and 7 show the probability Density Function of fibre delay line 1 for 6% CoS1 and 66% CoS1 respectively. **Partial flexibility** gives better performance, in terms of delay line 1 occupancy, than **total flexibility** in the 6% CoS1. This small percentage can not use all the bandwidth offered by λ_1 , but when the CoS1 percentage get large (66%), λ_1 is not enough sufficient to transport CoS1 and thus the associated delay line becomes saturated. Figure 8 confirms that, by showing the delay line 1 occupancy evolution versus the CoS1 percentage. Next, we show the remaining fibre delay lines performance associated to λ_2 , λ_3 and λ_4 . Figure 9 illustrates the delay line 2 occupancy in the **Partial flexibility** case. It is clearly shown by this plot that the delay line 2 is almost saturated for 6%, 11% and 15% CoS1. This is due to the high corresponding percentage of CoS2 and CoS3 traffic which can not access to the bandwidth offered by λ_1 .

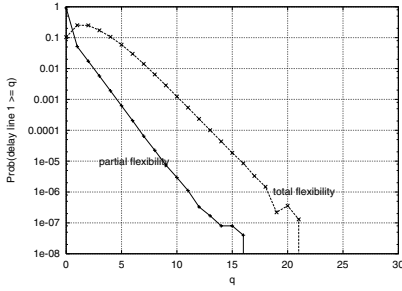


Fig. 6. Probability Density Function of one output fibre delay line 1 : 6% CoS1 traffic

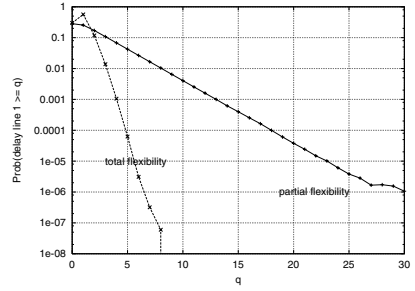


Fig. 7. Probability Density Function of one output fibre delay line 1 : 66% CoS1 traffic

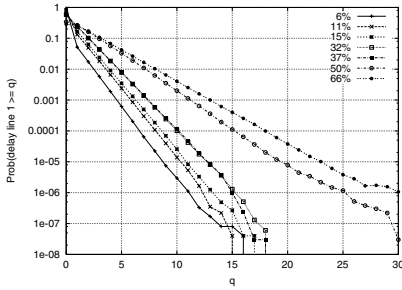


Fig. 8. Probability Density Function of one output fibre delay line 1 in the case of partial flexibility

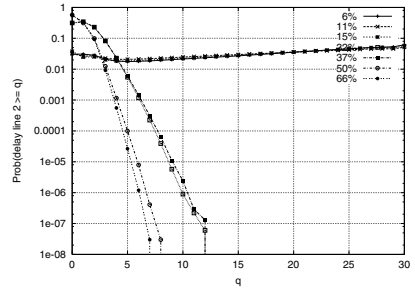


Fig. 9. Probability Density Function of one output fibre delay line 2 in the case of partial flexibility

Figure 10 shows optical packet fill ratio versus CoS1 traffic percentage obtained by the both resource allocation mechanisms considered. **Partial flexibility** gives better utilisation comparing with **total flexibility** in all cases, especially when CoS1 percentage becomes large ($>66\%$). This can be explained by the different electronic packet sizes associated to each CoS. By increasing CoS1 percentage, we increase the number of small packets (160 Bytes), and thus the optical slots fill ratio is more efficient.

5 Conclusion

We consider in this paper a multiservice-optical network where optics provide the underlying network architecture with a QoS requested by client layers at the periphery of optical network. We focus on end-to-end traffic management between the client layer and the ROM network in the presence of different traffic QoS requirements. Due to the lack of optical memories, the key idea is then to exploit, in an optimal way, the electronic memories in the electronic interface at the edge nodes of our optical network. Results have shown that shaping of

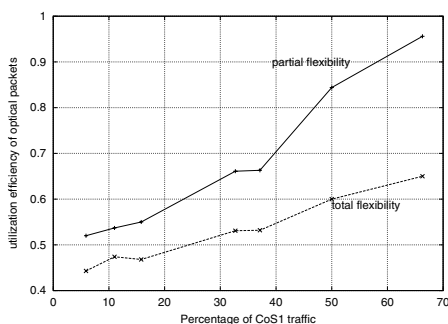


Fig. 10. Optical packet utilisation efficiency Vs CoS1 traffic percentage

data traffic eases the contention resolution in the core nodes by reducing the burstiness level of the traffic profile and by enabling the exploitation of optical resources, in terms of time, space and spectral techniques, in an efficient manner. There is clearly a trade-off to analyse between shaping parameters and delay line sizes to reach a target performance in terms of loss and delay. By simulation, we have obtained suitable values of fibre delay lines capacities using shaping mechanism at the periphery of ROM network. We have also studied the efficiency of optical packet utilisation by comparing two optical resource allocation mechanisms, namely total flexibility and partial flexibility.

References

1. Le Sauze, N. and al.: Optical packet vs circuit switching in Metropolitan networks. ECOC'99, vol.1,p14-15, Nice, France, September 1999.
2. Gravey, P. and al.: Multiservice Optical Network: Main concepts and First achievements of the ROM program. Journal of Lightwave Technology, January 2001.
3. Chiaroni, D. and al.: Experimental validation of an all-optical network based on 160 Gbit/s throughput packet switching nodes. ECOC'98, Edinbourg, UK, September 1997.
4. Elbiaze, H. and al.: Impact of shaping on network performance with self-similar traffic'. PROC of ICT'2000, Acapulco, Mexico, May 2000.
5. Leland, W. E. and al.: On the self-similar nature of Ethernet traffic. IEEE/ACM trans. Net, vol. 2, no.1, pp, 1-15, 1994.
6. Neilson, D. T. et al.: Fully provisioned 112x112 micro-mechanical optical crossconnect with 35.8 Tb/s demonstrated capacity. OFC 2000, PD12, Baltimore, USA, March 2000.

Performance Comparison of Wavelength Routing Optical Networks with Chordal Ring and Mesh-Torus Topologies

Mário M. Freire¹ and Henrique J.A. da Silva²

¹ Department of Computer Science, University of Beira Interior
Rua Marquês d'Ávila e Bolama, 6200-001 Covilhã, Portugal
mario@noe.ubi.pt

² Department of Electrical Engineering, University of Coimbra - Site II
3030-290 Coimbra, Portugal
hjas@dee.uc.pt

Abstract. In this paper, we present a comparison of the blocking performance in wavelength routed optical networks with chordal ring and mesh-torus topologies. The comparison is focused on chordal rings with a chord length of $\sqrt{N} + 3$, being N the number of nodes, since this chord length leads to the smallest network diameter. This performance comparison revealed an important feature of chordal rings: very small blocking gains were observed, due to the increase of the node degree from 3 (chordal ring with a chord length of $\sqrt{N} + 3$) to 4 (mesh-torus). The comparison is made for networks with 100 and 1600 nodes. The influence of wavelength interchange on these small gains is also investigated: the node degree gain is very small and increases slightly as the converter density increases. Thus, if a small blocking performance degradation is allowed, the choice of a chordal ring with a chord length of $\sqrt{N} + 3$, instead of a mesh-torus, leads to a reduction in the number of network links, and hence in the total cable length, since the number of links in a N -node chordal ring is $3N$, and the number of links in a N -node mesh-torus is $4N$.

1 Introduction

IP (Internet Protocol) networks based on WDM (Wavelength Division Multiplexing) are expected to offer an infrastructure for the next generation Internet [1]-[2]. However, up to now, WDM has been used to satisfy the bandwidth requirements imposed by the traffic growth. Actually, the worldwide deployment of WDM systems is seen as the first phase of optical networking. After this phase, the introduction of optical add/drop multiplexers in a linear architecture and the use of WDM protection switches are expected. This architecture will rapidly evolve to a WDM ring architecture, and a further possible evolution scenario may be the interconnection of WDM rings and mesh networks [3]. Whereas the evolution from the point-to-point

WDM transmission system to interconnected rings is clear from a physical topology point of view, the optimal topology to be used for the mesh network is a subject less studied. In [4], a study is presented of the influence of node degree on the fiber length, capacity utilization, and average and maximum path lengths of wavelength routed mesh networks. It is shown that average node degrees varying between 3 and 4.5 are of particular interest.

Here, we consider chordal rings, which are a particular family of regular graphs of degree 3 [5]. A chordal ring is basically a ring network, in which each node has an additional link, called a chord. The number of nodes in a chordal ring is assumed to be even, and nodes are indexed 0, 1, 2, ..., $N-1$ around the N -node ring. It is also assumed that each odd-numbered node i ($i=1, 3, \dots, N-1$) is connected to a node $(i+w) \bmod N$, where w is the chord length, which is assumed to be positive odd and, without loss of generality, we also assume that $w \leq N/2$, as in [5]. In this paper, we compare the blocking performance of chordal ring networks with a chord length of $\sqrt{N} + 3$, being N the number of nodes, with the performance of mesh-torus networks which have a node degree of 4.

For a given number of nodes N , different chordal rings can be obtained by changing the chord length. Fig. 1 shows a chordal ring with $N=20$ and $w=7$. Fig. 2 shows a mesh-torus network with 16 nodes.

The remainder of this paper is organized as follows. The analytical model used to compute the blocking probability in wavelength routed optical networks is briefly described in section 2. The performance comparison of chordal ring and mesh-torus networks is presented in section 3. Main conclusions are presented in section 4.

2 Evaluation of Blocking Probability

To compute the blocking probability in optical networks with wavelength interchange, we have used the model given in [6], since it applies to ring topologies, has a moderate computational complexity, and takes into account dynamic traffic and the correlation between the wavelengths used on successive links of a multi-link path. Moreover, this model is suitable for the study of the influence of wavelength interchange on the network performance.

The following assumptions are used in the model [6]: 1) Call requests arrive at each node according to a Poisson process with rate λ , with each call equally likely to be destined to any of the remaining nodes; 2) Call holding time is exponentially distributed with mean $1/\mu$, and the offered load per node is $\rho=\lambda/\mu$; 3) The path used by a call is chosen according to a pre-specified criterion (e.g. random selection of a shortest path), and does not depend on the state of the links that make up a path; the call is blocked if the chosen path can not accommodate it; alternate path routing is not allowed; 4) The number of wavelengths per link, F , is the same on all links; each node is capable of transmitting and receiving on any of the F wavelengths; each call requires a full wavelength on each link it traverses; 5) Wavelengths are assigned to a session randomly from the set of free wavelengths on the associated path.

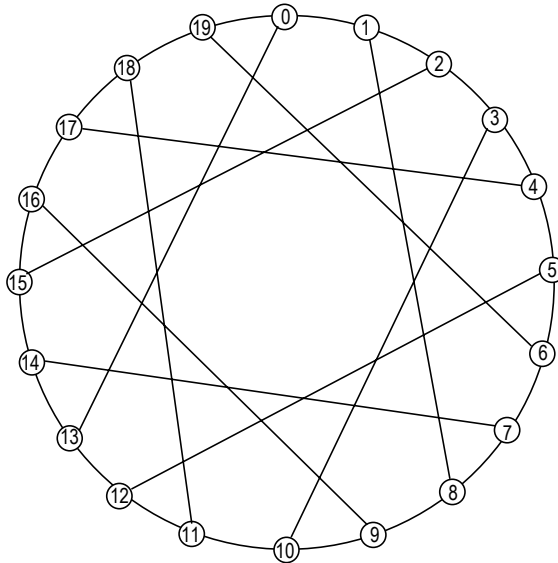


Fig. 1. Chordal ring network with $N=20$ nodes and $w=7$.

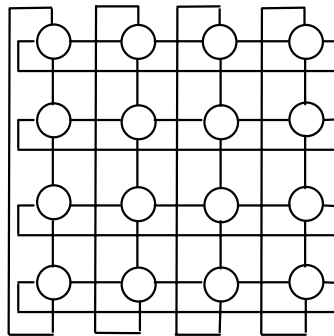


Fig. 2. Mesh-torus network with 16 nodes.

In addition to the above assumptions, it is assumed in [6] that, given the loads on links 1, 2, ..., $i-1$, the load on link i of a path depends only on the load on link $i-1$ (Markovian correlation model).

The analysis presented in [6] also assumes that the hop-length distribution is known, as well as the arrival rates of calls at a link that continue, and those that do not, to the next link of a path. The call arrival rates at links have been estimated from the arrival rates of calls to nodes, as in [6]. The hop-length distribution is a function of the network topology and the routing algorithm, and is easily determined for most regular topologies with the shortest-path algorithm. For the bi-directional N -node mesh-torus network, the hop-length distribution was easy to find. However, for the bi-directional N -node chordal ring with chord length w , it was not possible to obtain a general expression for the hop-length distribution. We have found analytical

expressions for the hop-length distribution when the chord length is 3 ($w=3$), when the chord length is maximum ($w=N/2$ or $w=N/2-1$), when the chord length is as close as possible to the mean chord length ($w=N/4$), and for a chord length that leads to the smallest network diameter ($w=\sqrt{N}+3$). In this paper, we concentrate on the latter chord length ($w=\sqrt{N}+3$). For this chord length, the hop-length distribution, p_l , with $N=m^2$ and $m=10+2k$ ($k=0,1,2,3,\dots$), is given by:

$$p_l = \begin{cases} \frac{3l}{N-1}, & \text{for } 1 \leq l \leq \frac{m}{2} + 1 \\ \frac{2m+6-l}{N-1}, & \text{for } \frac{m}{2} + 2 \leq l \leq m-4 \quad \text{and } N \geq 144 \\ \frac{\frac{m}{2}+14}{N-1}, & \text{for } l = m-3 \\ \frac{13}{N-1}, & \text{for } l = m-2 \\ \frac{4}{N-1}, & \text{for } l = m-1 \end{cases} \quad (1)$$

It can be shown that the average hop length, $\bar{H}_c$, of a chordal ring with $N=m^2$ nodes, being $m=10+2k$ ($k=0,1,2,3,\dots$) and with a chord length of $w=\sqrt{N}+3$, is given by:

$$\bar{H}_c = \frac{N + 59\sqrt{N} - 144}{2N - 2} - \frac{-14Nm + 45N + 374\sqrt{N} - 288}{24N - 24} \quad (2)$$

For a N -node mesh-torus network, with $N = m^2$, $m \geq 4$ and m even, the hop-length distribution, p_l , is given by:

$$p_l = \begin{cases} \frac{4l}{N-1}, & \text{for } 1 \leq l \leq \frac{m}{2} - 1 \\ \frac{2m-2}{N-1}, & \text{for } l = \frac{m}{2} \\ \frac{4m-4l}{N-1}, & \text{for } \frac{m}{2} + 1 \leq l \leq m-1 \\ \frac{1}{N-1}, & \text{for } l = m \end{cases} \quad (3)$$

The average hop length, $\bar{H}_m$, for a N -node mesh-torus network, with $N = m^2$, $m \geq 4$ and m even, is given by:

$$\bar{H}_m = \frac{Nm}{2(N-1)} . \quad (4)$$

The performance assessment presented in this paper is based on the evaluation of the path blocking probability. However, a direct comparison of blocking probabilities is sometimes preferred in order to put in evidence some features and to quantify benefits. Besides the blocking probability, we consider the blocking gain due to an increase in the node degree (G_{nd}). Since we consider only blocking gains along the paper, G_{nd} is also called here as the node degree gain. This metric is defined as:

$$G_{nd} = \frac{P_{(D-1)}}{P_D} , \quad (5)$$

where $P_{(D-1)}$ is the blocking probability in a network with a node degree $D-1$, and P_D is the blocking probability in a network with a node degree D (both obtained for the same number of nodes, wavelengths per link, and load per node).

3 Performance Comparison

In this section we compare the blocking performance of mesh-torus networks and of chordal ring networks with a chord length of $\sqrt{N} + 3$, being N the number of nodes.

Fig. 3 shows the blocking gain due to an increase of the node degree from 3 to 4, as a function of the load per node, for chordal ring and mesh-torus networks, both with 100 nodes and without wavelength interchange. For $w=N/4$, gains of the order of 10^2 and 10^5 were obtained for 8 and 16 wavelengths per link, respectively, with a load per node of 0.1 Erlang. For the case of $w=\sqrt{N} + 3$, very small gains were observed. As the load per node decreases from 5 Erlang to 0.01 Erlang, the variation of the node degree gain remains within one order of magnitude for the numbers of wavelengths per link considered: 4, 8, 12 and 16. In the following we consider chordal rings with chord lengths of only $\sqrt{N} + 3$.

Fig. 4 shows the blocking probability versus converter density for chordal ring and mesh-torus networks, both with 100 nodes and a load per node of 0.5 Erlang. As can be seen, the blocking probabilities in both cases are close. The corresponding node degree gain for 12 wavelengths per link is depicted in Fig. 5. From this figure, we may observe that the node degree gain slightly increases as the converter density increases. Besides, the node degree gain is very small.

We have further increased the number of nodes to 1600. Fig. 6 shows the blocking probability versus load per node for chordal ring and mesh-torus networks, both with 1600 nodes and without wavelength interchange. In this case, as the load per node decreases from 0.1 Erlang to 0.001 Erlang, the variation of the node degree gain remains within one order of magnitude for the numbers of wavelengths per link that we

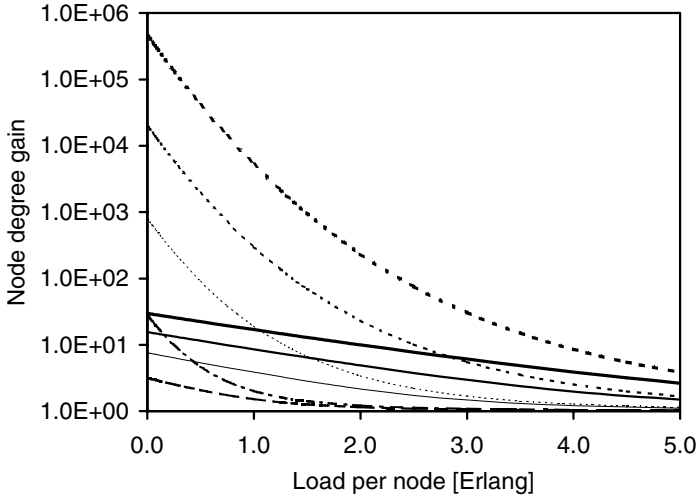


Fig. 3. Blocking gain due to an increase of the node degree from 3 (chordal ring networks with 100 nodes) to 4 (mesh-torus networks with 100 nodes), for chord lengths of $N/4$ and $\sqrt{N} + 3$ and without wavelength interchange. $w=N/4$, $F=4$: - - - -; $w=N/4$, $F=8$: ·····; $w=N/4$, $F=12$: ····; $w=N/4$, $F=16$: - · - ·; $w=\sqrt{N} + 3$, $F=4$: - - -; $w=\sqrt{N} + 3$, $F=8$: —; $w=\sqrt{N} + 3$, $F=12$: —; $w=\sqrt{N} + 3$, $F=16$: —.

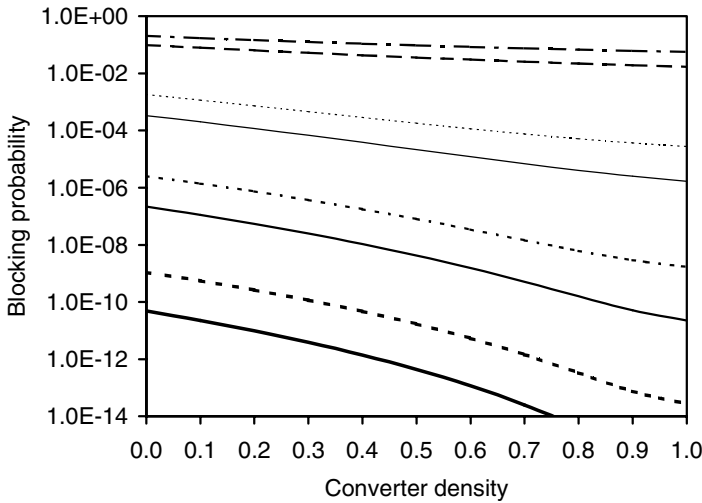


Fig. 4. Blocking probabilities for chordal rings with $w=\sqrt{N} + 3$ and mesh-torus networks, both with 100 nodes and a load per node of 0.5 Erlang. Chordal ring with $F=4$: - - - -; Chordal ring with $F=8$: ·····; Chordal ring with $F=12$: ····; Chordal ring with $F=16$: - · - ·; Mesh-torus with $F=4$: - - -; Mesh-torus with $F=8$: —; Mesh-torus with $F=12$: —; Mesh-torus with $F=16$: —.

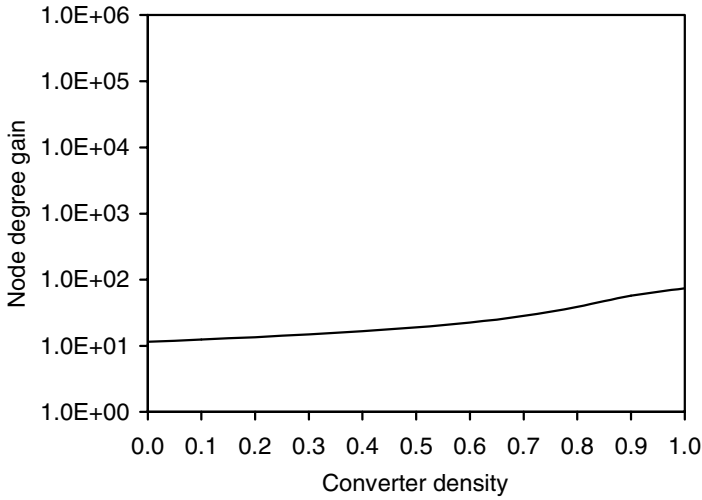


Fig. 5. Blocking gain due to an increase of the node degree from 3 to 4, as a function of the converter density, for chordal ring networks with $w=\sqrt{N}+3$ and mesh-torus networks, both with 100 nodes, 12 wavelengths per link and a load per node of 0.5 Erlang.

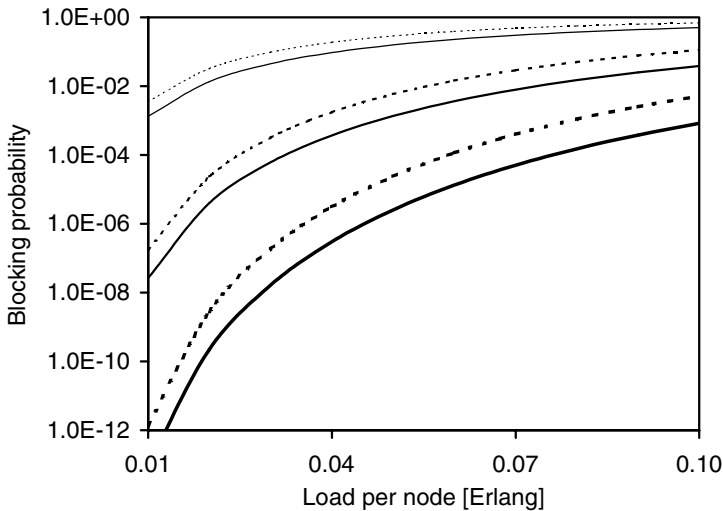


Fig. 6. Blocking probabilities for chordal ring networks with $w=\sqrt{N}+3$ and mesh-torus networks, both with 1600 nodes and without wavelength interchange. Chordal ring with $F=4$: $\cdots\cdots$; Chordal ring with $F=8$: $-\cdot-\cdot-$; Chordal ring with $F=12$: $- - - -$; Mesh-torus with $F=4$: — ; Mesh-torus with $F=8$: — ; Mesh-torus with $F=12$: — .

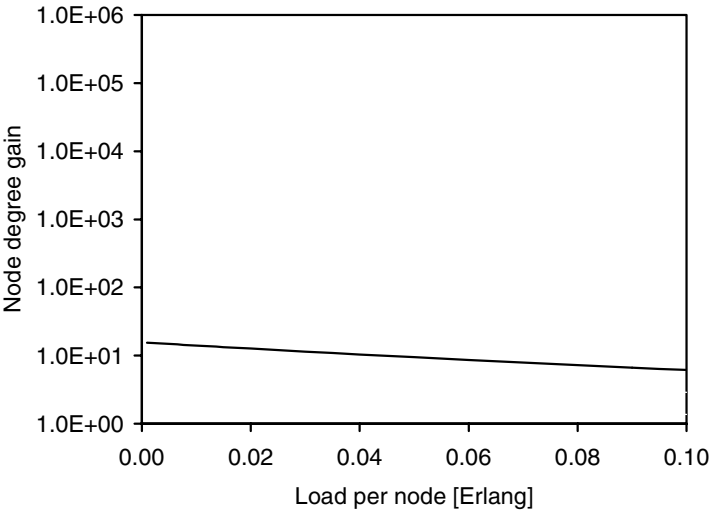


Fig. 7. Blocking gain due to an increase of the node degree from 3 to 4, as a function of the load per node, for chordal ring networks with $w=\sqrt{N}+3$ and mesh-torus networks, both with 1600 nodes and 12 wavelengths per link without interchange.

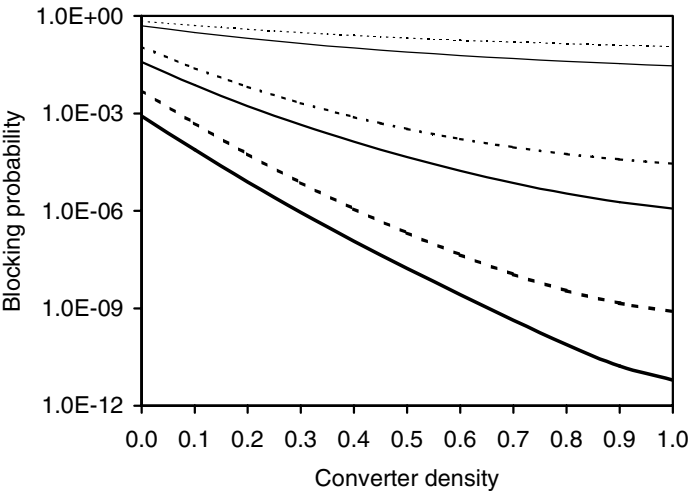


Fig. 8. Blocking probability versus converter density for chordal ring networks with $w=\sqrt{N}+3$ and mesh-torus networks, both with 1600 nodes and a load per node of 0.1 Erlang. Chordal ring with $F=4$:; Chordal ring with $F=8$: - . - .; Chordal ring with $F=12$: - - - -; Mesh-torus with $F=4$: —; Mesh-torus with $F=8$: — — —; Mesh-torus with $F=12$: — — — —.

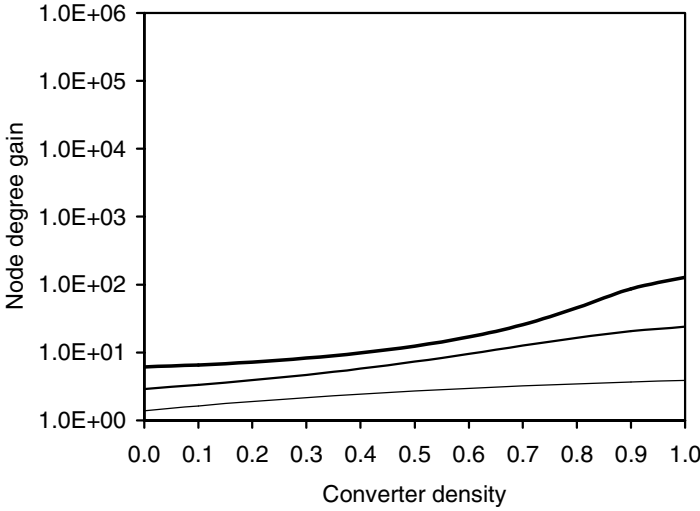


Fig. 9. Blocking gain due to an increase of the node degree from 3 to 4, as a function of the converter density, for chordal ring networks with $w=\sqrt{N}+3$ and mesh-torus networks, both with 1600 nodes and a load per node of 0.1 Erlang. $F=4$: —; $F=8$: - - -; $F=12$: —.

have considered: 4, 8 and 12. See also Fig. 7 for the case of 12 wavelengths, where the difference between both curves is higher. Fig. 8 shows the influence of converter density on the blocking probability for mesh-torus and chordal rings with $w=\sqrt{N}+3$, both with 1600 nodes and a load per node of 0.1 Erlang. In both networks, wavelength interchange is more helpful as the number of wavelengths per link increases, but, even for the case of 12 wavelengths per link, the node degree gain increases from 6.1 to only 127.9, as the converter density increases from 0 to 1 (see Fig. 9). Again, the node degree gain is very small and increases slightly as the converter density increases.

These very small node degree gains, observed as we increase the node degree from 3 (chordal rings with $w=\sqrt{N}+3$) to 4 (mesh-torus networks), may be explained by the dependence of p_l (hop length distribution) with l in both cases. The average hop-length is of the order of m , $O(m)$, in both cases (see equations 2 and 4).

If a small blocking performance degradation is allowed, the choice of chordal ring networks with $w=\sqrt{N}+3$, instead of mesh-torus networks, leads to a reduction in the number of network links, and hence in the total cable length, since the number of links in a N -node chordal ring is $3N$, and the number of links in a N -node mesh-torus is $4N$. However, there are some restrictions that limit the practical implementation of chordal rings with $w=\sqrt{N}+3$ (as well as mesh-torus), when compared with other chord lengths. In fact, the smallest network diameter was obtained with $w=\sqrt{N}+3$ for a N -node chordal ring, where N is a square ($N=m^2$) and $N \geq 64$. The restriction

associated with the square is not imposed to other chord lengths such as $w=N/4$ or $w=3$.

4 Conclusions

We have presented a performance comparison of wavelength routing optical networks with chordal ring and mesh-torus topologies. This comparison revealed an important feature of chordal rings: the performance of chordal ring networks, with chord length of $\sqrt{N}+3$, is similar to the performance of mesh-torus networks. For this comparison, networks with 100 and 1600 nodes have been considered. Concerning the influence of wavelength interchange on the small node degree gains, it was shown that the node degree gain is very small and increases slightly as the converter density increases.

Since the performance of mesh-torus and chordal rings, with chord length of $\sqrt{N}+3$, is similar, the choice of a chordal ring with a chord length of $w=\sqrt{N}+3$, instead of a mesh-torus network, leads to a reduction in the number of network links, and hence in the total cable length, since the number of links in a N -node chordal ring is $3N$, and the number of links in a N -node mesh-torus is $4N$.

Acknowledgements

Part of this work has been supported by Fundação para a Ciência e Tecnologia, Portugal, in the framework of project TRANSPARENT (TRANSPort, Access and distRibution in optically multiplEXed NeTworks).

References

1. Listanti, M., Sabella, R. (eds): Optical Networking Solutions for Next-Generation Internet Networks. *IEEE Communications Magazine* **38** (2000) 79-122
2. Su, D. H., Griffith, D. A.: Standards: Standards Activities for MPLS Over WDM Networks. *Optical Networks Magazine* **1** (2000) 3/66-69
3. Arijs, P., Caenegem, B., Demester, P., Lagasse, P., Parys, W., Achten, P.: Design of Ring and Mesh Based WDM Transport Networks. *Optical Networks Magazine* **1** (2000) 3/25-40
4. Hjelme, D. R.: Importance of Meshing Degree on Hardware Requirements and Capacity Utilization in Wavelength Routed Optical Networks. In Gagnaire, M., van As, H. R. (eds.): *Proceedings of the 3rd IFIP Working Conference on Optical Network Design and Modeling (ONDM'99)*. Paris (1999) 417-424
5. Arden, B. W., Lee, H.: Analysis of Chordal Ring Network. *IEEE Transactions on Computers* **C-30** (1981) 291-295
6. Subramaniam, S., Azizoglu, M., Somani, A. K.: All-Optical Networks with Sparce Wavelength Conversion. *IEEE/ACM Transactions on Networking* **4** (1996) 544-557

Achieving End-to-End Throughput Guarantee for Individual TCP Flows in a Differentiated Services Network

X. He and H. Che

Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802, USA

Abstract. The differentiated services (DS) architecture provides Quality of Service(QoS) assurance to different classes of service (CoSs). Our previous research results show that both intradomain and interdomain best-effort traffic can have adverse impact on the interdomain TCP assured service traffic. With the Measurement-based connection-oriented assured service model we developed in our previous research, we are able to provide end-to-end TCP throughput assurance for each CoS. However, for each TCP session within a CoS, the throughput may not be able to be guaranteed. In this paper, we propose modified marking and dropping policy based on our previous research results. The simulations show that with these techniques, the end-to-end throughput for each individual TCP flow can be significantly improved. It also maintains the high scalability of the DS architecture without requiring the core router to keep flow per-flow state information.

1 Introduction

As the Internet evolves into a global commercial infrastructure, there is a growing need to support quality of service (QoS) to applications. Recently, a radical approach, known as differentiated services (DS) [1,5], has attracted much attention. The DS model is based on the assumption that resources are *abundant* in the core and bottlenecks occur only at the border nodes between domains. While offering multiple CoSs, the DS model ensures scalability by keeping a stateless core and adhering to the IP (i.e. Internetworking Protocol) hop-by-hop forwarding paradigm. However, a key problem for this model is the conflict between maximizing resource utilization and achieving a high service assurance. In order to provide high service assurance, enough resources need to be provisioned to all the possible paths in the direction from a source to a destination.

In our previous papers, [9], we developed a Measurement based Connection-Oriented Assured Service Model (MCOAS). With MCOAS, a high end-to-end service assurance can be achieved for the TCP traffic of assured service (AS), while a reasonably high throughput for best-effort traffic is also maintained. However, there is an issue requiring further study. In the MCOAS model, the

fairness problem exists between each TCP flow within the aggregated flow. In other words, in the MCOAS model, the end-to-end throughput assurance is provided *aggregately* to the AS traffic. I. Yeom and A. Reddy observed the unfair issue, and improved the model to achieve better service assurance and fairness[7,8,2]. However, these works assumes the source will always send out data as fast as they can, or requires the edge devices be able to notify the sender to slow down. In order to solve these problems, in this paper, we propose new marking, measuring and dropping algorithms which are able to solve the fairness problem based on the MCOAS model. This paper heavily relies on our previous research work[9].

The rest of the paper is organized as follows. Section 2 presents a background introduction on MCOAS model and our research motivation. Section 3 describes the proposed schemes including marking and dropping algorithms. Section 4 gives the experiment results on the proposed scheme. Finally, Section 5 concludes the paper and presents future research directions.

2 Background

The original idea for designing AS for TCP applications was proposed by Clark and Fang [6]. Each AS session using TCP receives a guaranteed minimum bandwidth called target rate. Traffic is policed at every Internet service provider (ISP) domain edge node. At the edge node, conformant packets are marked as IN-profile and non-conformant packets are marked as OUT-profile. Both IN and OUT packets are injected into the core of the network, and the OUT packets are treated the same way as the best-effort packets. In each core router, a single first-in- first-out (FIFO) queue is used for both AS and best-effort traffic. A 2-level RED (i.e. random early detection) [10,11] packet dropping algorithm, called RIO (RED in-and-out) [6], is run based on traffic type. At each and every domain boundary, traffic is policed locally, and packets are subject to remarking before being injected into another domain, based on local congestion situation. However, since there is no end-to-end resource provisioning, an end-to-end service assurance is not guaranteed. Several other works on the improvement of this model in an attempt to achieve better service assurance and fairness were proposed [7,8,2], all based on a connectionless forwarding paradigm.

The above studies did not consider end-to-end performance of the AS TCP sessions, in the presence of possible the cross best-effort traffic with small round-trip time (RTT). The cross traffic could occur within a domain or at a domain boundary. A question is whether local control at each domain boundary can guarantee end-to-end performance for AS flows that cross multiple domains. To answer this question, we did simulation tests on the approach proposed in [6], using NS-2 from LBNL (Lawrence Berkerley National Laboratory).

The network setup is shown in Fig. 1. It is composed of three domains. The link bandwidth between routers are 33 *Mbps* each and the buffer size is 50 packets for each output port of a router. Each hosts in domain 1 has an AS TCP

session with one of the hosts in domain 3. The aggregated target rate of these 10 hosts for AS is 30 *Mbps*.

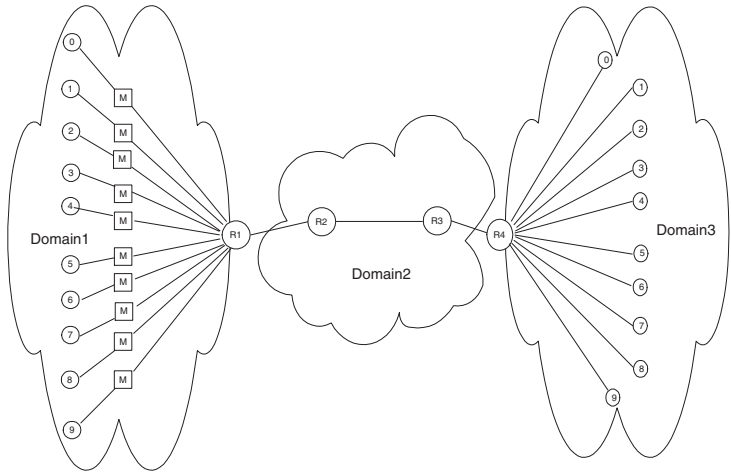


Fig. 1. Network setup for experiment 1

Table 1. Aggregated Throughputs/Goodputs of Assured Services for experiments 1 and 2

Flow	RTT(ms)	Target	Test1	Test2
0	20	5Mbps	5.2/5.1	3.6/3.5
1	20	1Mbps	3.0/2.8	1.0/0.9
2	40	5Mbps	4.1/3.9	2.8/2.7
3	40	1Mbps	2.0/1.9	1.0/0.9
4	50	5Mbps	4.0/4.0	3.0/3.0
5	50	1Mbps	2.1/1.9	0.8/0.8
6	70	5Mbps	3.7/3.6	2.6/2.6
7	70	1Mbps	1.6/1.5	0.8/0.8
8	100	5Mbps	3.5/3.4	2.6/2.6
9	100	1Mbps	1.2/1.2	0.7/0.6
Total		30Mbps	30.3/29.4	18.9/18.3

The parameters for RIO are set at $(min_{in}, max_{in}, P_{in}) = (40, 70, 0.02)$ for IN packets and $(min_{out}, max_{out}, P_{out}) = (10, 30, 0.2)$ for OUT packets. For more details on RIO, please refer to [6].

Two experiments were performed. In the first experiment, we assumed that there is no cross traffic. In the second experiment, we added 50 best-effort TCP flows between nodes R2 and R3 with a RTT of 10 *ms* each, representing the

intradomain cross traffic (we can also interpret it as the interdomain cross traffic coming into domain 2 via R2 and going out to other domains via R3).

Table 1. summarizes the results for the two experiments. Here we focus on the aggregated throughput and goodput. As one can see, RIO achieves rather high aggregated service assurance. However, the situation becomes quite different in the presence of the cross best-effort traffic. Most of the AS sessions fall short of their target rates. Even worse, the achieved aggregated throughput/goodput are only about two third of the target value.

In our previous paper, we developed a MCOAS model [9] which can provide end-to-end TCP throughput assurance. The MCOAS model is a connection oriented service model composed of a series of measures which includes: (a) a path pinning mechanism for AS allowing aggregated bandwidth reservation for AS at *each intermediate router* in the forwarding path; (b) a packet marking strategy; (c) a dropping policy; (d) an adaptive dropping-threshold calculation method for queue management based on aggregated reserved bandwidth and real-time traffic measurement.

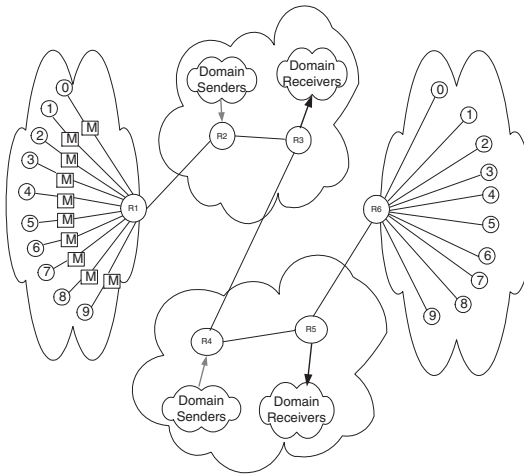


Fig. 2. Aggregated throughputs/goodputs for MCOAS and RIO

To show the performance of MCOAS, we consider a network setup with one more domain in the data path as shown in Fig. 2. In this experiment, There are 10 AS TCP sessions between the hosts in domain 1 and domain 4. Their target rates and RTTs are listed in Table 1. The aggregated target rate is 30 *Mbps*. The link capacities between R1 and R2, R2 and R3, R3 and R4, and R5 and R6 are all 33 *Mbps*. The link capacity between R4 and R5 is 50 *Mbps*. There are 30 cross best-effort flows from R2 to R3 in domain 2 and 30 cross best-effort flows from R5 to R6 in domain 3. The simulation is performed for MCOAS with $n = 2$ and RIO. Both throughput and goodput are measured. This time we want to

test the performance of each session and the results are listed in Table 2, with a slash separating the throughput from the goodput.

Table 2. Individual throughput/goodput for MCOAS and RIO

Flow	RTT(ms)	Target	RIO	MCOAS
0	20	5	3.4/3.3	4.7/4.5
1	20	1	0.9/0.8	2.7/2.6
2	40	5	3.2/3.1	4.4/4.3
3	40	1	0.6/0.6	2.0/1.9
4	50	5	2.9/2.8	4.0/4.0
5	50	1	0.8/0.7	2.0/1.8
6	70	5	2.4/2.4	3.8/3.7
7	70	1	0.8/0.7	1.5/1.4
8	100	5	2.8/2.8	3.0/3.0
9	100	1	0.5/0.5	1.3/1.2
Total		30	18.1/17.6	29.2/28.4

Table 3. Aggregated throughputs/goodputs of best-effort traffic for MCOAS and RIO, and the total throughputs/goodputs for MCOAS in domain 2 and domain 3

	In Dmain 2	In Domain 3
RIO	14.40/12.38	29.25/27.46
MCOAS	1.74 /1.51	19.30/17.48
Total Rate with RIO	32.53/30.00	47.38/45.08
Total Rate with MCOAS	30.95/29.93	48.51/45.90

As one can see, MCOAS outperforms RIO for all the AS sessions and again, it offers superior performance to RIO in terms of aggregated throughput guarantee.

To see the performance of the best-effort traffic, Table 3 lists the aggregated throughput/goodput for the cross best-effort traffic in both domains. As expected, MCOAS gracefully suppresses the best-effort traffic in both domain 2 and domain 3, offering rather high goodputs at about 1.5 *Mbps* and 17.5 *Mbps* in the respective domains. Hence, MCOAS can locally suppress cross traffic, resulting in a near-optimal global resource utilization. For the detailed information about the MCOAS model, please refer to [9].

However, an open issue in the MCOAS model is to how to solve the fairness issue between the individual flows. From Table. 2, we can see that the flow with lower target rate achieves higher throughput than it requires while the flow with higher target rate achieves less throughput then it requires. Since the ultimate target is to provide per-flow QoS assurance to each users, it is necessary for us to solve this fairness problem.

3 Proposed Scheme

In this section, we propose our solutions based on our MCOAS model to solve the fairness problems.

3.1 Marking Policy

In the MCOAS, the packets from AS are marked as AS or EX packets. The EX packets from the all the AS flows are treated in the same way. However, drop EX packets from different different TCP session in fact has different impact on the throughput.

Figure. 3 illustrates the different impact. In the figure, “x” representees a packets drop. Assume that both TCP sessions receive a packets drop at the same time, it is clear that the average throughput of the session with higher target rate will not meet the target rate while the session with lower target rate will meet the rate. It is clear drop the EX packet from the session with high target rate will have more significant impact. Notice that in this example, the EX packets in the queue are primarily from the low target rate session, the queue is primarily occupied by the EX packets from the low target rate session which increases the probability of dropping a packet from the high target rate session. This causes the fairness issue we have observed. In order to solve this problem, we propose a modified marking policy.

In RIO and MCOAS presented in the previous sections, the sending rate is measured with the Time Sliding Window (TSW) algorithm which is described with the following equation

Initially:

Win_ Length=a constant;
 $R_{tsw} = \text{connection's target rate, } R_{target};$
T_ front=0

Upon each packet arrival, TSW updates its state variables as follows:

Bytes_ in_ TSW= $R_{tsw} * \text{Win_ length};$
New_ bytes = **Bytes_ in_ TSW** + **pkt_ size**;
 $R_{tsw} = \text{New_ bytes} / (\text{now} - \text{T_ front} + \text{Win_ length});$
T_ front = **now**

Whereas, **now** is the time of current packet arrival, and **pkt_ size** is the packet size of the arriving packet.

For the details, please refer to [6]. The TSW algorithm only measures the average with in the fixed-length time window. Since the window length is usually chosen to be short, the sending rate measured by the TSW algorithm can be considered at the real time sending speed.

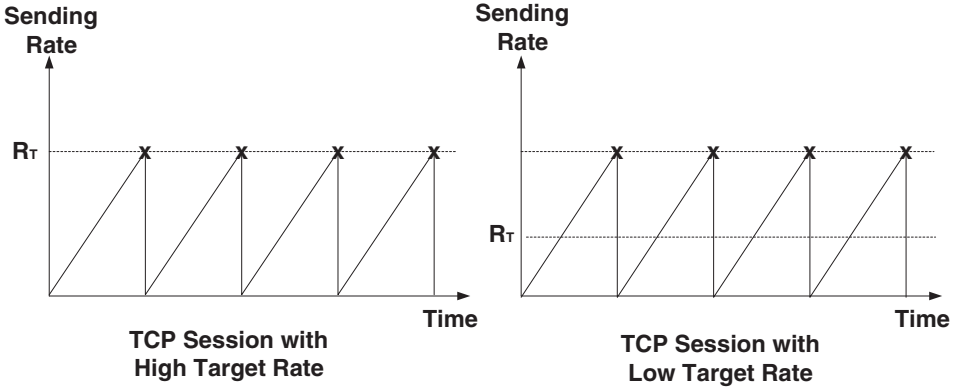


Fig. 3. MCOAS performance with on-off AS sessions

In the new marking policy, two sending rates are measured: 1) we measure the sending rate, R_{tsw} , with Time TSW algorithm as usual. 2) we also measure the average session sending rate, $R_{session}$, of each TCP session which can be calculated with the following equation

$$R_{session} = \frac{N_{sender}}{T_{duration}} \quad (1)$$

Whereas, N_{sender} the total bits are sent by the sender during the current session; and $T_{duration}$ is the duration of the current session.

Since MOCAS model is a connection-oriented model and the establishment and teardown process are required, the $R_{session}$ can be easily measured. Accordingly, when the $R_{tsw} < R_{target}$ and $R_{session} < R_{target}$, the packet will be marking as *AS* packet. When $R_{tsw} > R_{target}$ and $R_{session} < R_{target}$, we mark the packet as *MA* packet. When the $R_{session} > R_{target}$, we mark the packet as *EX* packet.

3.2 Dropping Algorithm

The new dropping policy is described as follows

If (*Packet Type is AS*)

Process the packet in the same way as RIO processes an IN packet

Else if (*packet Type is BE and queue length of the best-effort packets* $> K_{be}$)

Drop the BE packet

Else if *Packet Type is MA*

Process the packet in the same way as RIO processes an IN packet with different thresholds

Else

Treat EX and BE packets the same way as OUT in RIO

The only difference between this policy and MCOAS is that in this policy, we use a three parameter sets to further discriminates between the AS, MA and EX packets.

4 Simulation Results and Analysis

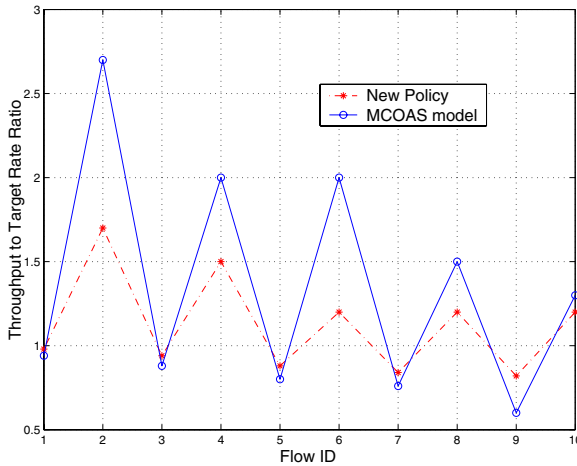


Fig. 4. MCOAS performance with on-off AS sessions

To examine the performance of new proposed schemes, we run the simulation with the network setup in Fig. 2 again. This time we want to test the performance of each session and the results are listed in Table 4, with a slash separating the throughput from the goodput.

The parameters for dropping policy are set at $(min_{in}, max_{in}, P_{in}) = (40, 70, 0.02)$ for IN packets, $(min_{ma}, max_{ma}, P_{ma}) = (30, 50, 0.08)$ for MA packets and $(min_{out}, max_{out}, P_{out}) = (10, 30, 0.2)$ for BE and EX packets.

From Table 4, it is clear that with the modifications, the performance of each TCP session has been significantly improved. In order to show the improvement, we plot the throughput deviation for both policies in Fig. 3. The solid line represents the end-to-end throughput to the target target ratio. And the dotted line shows the same ratio of new policy. It is clear that with the new policy, the deviation of each TCP session is significantly reduced.

With the new policies, the MCOAS model maintains high end-to-end aggregated throughput. However, when we look at the throughput achieved by each TCP session, it is clear that with the new policies, the throughput achieved by each session is close to the ideal throughput. However, the MCOAS model still can not provide the ideal throughput guarantee to each TCP session.

Table 4. Individual throughput/goodput for the proposed scheme and MCOAS (UINT:Mbps)

Flow	RTT(ms)	Target	New Policy	MCOAS
0	20	5	4.9/4.8	4.7/4.5
1	20	1	1.7/1.6	2.7/2.6
2	40	5	4.7/4.5	4.4/4.3
3	40	1	1.5/1.4	2.0/1.9
4	50	5	4.4/4.3	4.0/4.0
5	50	1	1.2/1.1	2.0/1.8
6	70	5	4.2/4.1	3.8/3.7
7	70	1	1.2/1.0	1.5/1.4
8	100	5	4.1/4.0	3.0/3.0
9	100	1	1.2/1.0	1.3/1.2
Total		30	29.1/27.8	29.2/28.4

5 Conclusions and Future Work

In this paper, new marking and dropping policy are proposed. We are able to show that based on the scheme proposed scheme, the fairness problem between individual flows within the aggregated flow can be solved. The proposed scheme will not require the core routers to keep the per-flow information so that it is highly scalable. When work with the MCOAS model, we are able to provide end-to-end throughput assurance to individual without requiring the core router to keep per-flow information.

It is worth mentioning that the MCOAS model is a connection-oriented model so it can be easily used on top of a MPLS enabled network [3]. The future work focus on how to combine these two technologies together.

References

1. D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC2475, Dec., 1998, and the references therein.
2. I. Yeom and A. Reddy, "Realizing throughput guarantees in a differentiated services network," *Proc. of IWQoS*, June 1999.
3. T. Li and Y. Rekhter, "Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)," IETF Draft: draft-li-paste-01.txt, September 1998.
4. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS," IETF RFC 2702, September 1999.
5. Y. Bernet, J. Binder, S. Blake, M. Carlson, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, and W. Weiss, "A Framework for Differentiated Services," Internet Draft <draft-ietf-diffserv-framework-01.txt>, Oct., 1998, and the references therein.
6. D. Clark and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service," *IEEE/ACM Transactions on Networking*, Vol. 6, No. 4, p. 362, Aug. 1998.

7. I. Yeom and A. Reddy, "Impact of marking strategy on aggregated flows in a differentiated services network," *Proc. of IWQoS*, June 1999.
8. I. Yeom and A. Reddy, "Marking of QoS Improvement," <http://dropzone.tamu.edu/~ikjun/papers.html>.
9. X. He and H. Che, "Achieving End-to-end Throughput Guarantee for TCP Flows in a Differentiated Services Network," *Proc. of IEEE ICCCN'2000*, Oct 2000.
10. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, Vol. 1, No. 4, p. 397, Aug. 1993.
11. Ott, T.J., Lakshman, T.V., and Wong, L.H. "SRED: stabilized RED," *INFOCOM '99* Vol. 3, p. 1346, Aug. 1999.

Performance Evaluation of Integrated Services in Local Area Networks

Jürgen Ehrensberger

Swiss Federal Institute of Technology, Telecommunications Laboratory,
CH-1015 Lausanne, Switzerland,
Juergen.Ehrensberger@epfl.ch

Abstract. Integrated Services over Local Area Networks are treated. We describe the main functionalities of LAN devices that allow the realization of the Integrated Services model in such networks. Based on an admission control algorithm capable of providing queueing delay guarantees across a LAN, we evaluate the share of the network capacity that can be used by flows requiring QoS guarantees. We find that for LANs of realistic size, only a small percentage of the network capacity can be made available in order not to violate the QoS guarantees.

1 Introduction

The goal of the Integrated Services (IntServ) model is to provide a well-defined QoS to data flows along the entire transmission path between the senders and the receivers of the flows. Local Area Networks (LANs) generally constitute the to last hops to terminal devices that act as senders and receivers of data flows. Traditional LANs, often composed of different LAN technologies, do not allow for service differentiation due to the lack of both QoS supporting mechanisms in LAN technologies like Ethernet and QoS signaling for data flows across different technologies. As real-time applications will gain importance in future communications systems, there has been much interest for the provision of service differentiation in LANs. This has been made possible by evolving link layer technologies and the standardization of capabilities needed for service differentiation by the IEEE for the most important LAN technologies, defined within the project 802, including the (switched) Ethernet/IEEE 802.3 and (shared/switched) Token Ring/IEEE 802.5 networks. Shared Ethernet networks are not capable of guaranteeing transmission delay bounds and thus are not suitable for realizing the IntServ model.

Service differentiation is achieved by assigning priorities to packets and using priority scheduling at forwarding devices. The use of priorities for service differentiation has as consequence that different flows using the same priority cannot be distinguished inside a LAN and thus cannot be isolated one from another at forwarding devices. This so-called *aggregate scheduling* is very different from the treatment of IntServ flows at routers. In this article we study the influence of aggregate scheduling on the performance of such networks. The evaluations

are based on an admission control algorithm that limits the high-priority traffic such that delay bounds can be guaranteed. The algorithm is based on recently derived delay bound results for networks with aggregate scheduling [1].

2 IEEE 802 LAN Model

Let us consider a layer 2 domain which we define as a closed network region in which frames are forwarded based on layer 2 addressing, thus without employing layer 3 forwarding functionality. A layer 2 domain consist of *segments* that are separated by *bridges* or *switches*. A segment is a physical medium to which one, two, or more senders are connected. Examples of segments are:

- a Token Ring,
- a half duplex link between two stations,
- one direction of a full duplex link.

Bridges are forwarding devices that operate at layer 2 (L2) of the OSI reference model. As such, they are independent of the higher layer protocols used. They accept incoming frames, decide based on the information contained in the frames to which output ports the frames have to be forwarded, and transmit them over the selected ports. The forwarding decisions are based on physical layer addresses, as opposed to logical addresses used in layer 3 (L3) devices (routers). Due to their simple operation mode, bridges can forward frames at high speed. Switches are similar to bridges as they also interconnect LAN segments, forward frames based on physical addresses, and filter traffic. They are different, however, in that switches are high-speed devices that make forwarding decisions in hardware whereas bridges operate in software. Therefore, switches can serve more ports than bridges. In the following we will use the term bridge to refer to both types of devices.

Since bridges operate at the data link layer, they cannot identify individual data flows, which are distinguished by elements of the layer 3 packet headers. Per-flow queueing, policing, and reshaping, which are required for Controlled Load (CL) and Guaranteed Service (GS), thus cannot be implemented. Bridges, however, can provide capabilities allowing an approximation of these services that may be sufficient for most practical purposes. The characteristics of IEEE 802 bridges are standardized in order to assure the interoperability of different LAN technologies. The latest revision of this standard has introduced new capabilities of bridges that provide the base for service differentiation in IEEE 802 LANs.

Such bridges have up to 8 output queues per port. The default scheduling algorithm uses static priorities assigned to the output queues. The specification for LAN bridges allows *user priorities* to be assigned to frames. A frame is mapped to an output queue depending on its user priority. User priorities, multiple output queues, and appropriate scheduling algorithms together make it possible to define different traffic classes, e.g., for best effort, video, and voice traffic. A traffic class is hence an aggregation of data flows which are given a similar service within a IEEE 802 network. The isolation of flows required for

Table 1. Service mapping of user priorities

User priority	Service
7	Network control
6	Delay sensitive appl., 10 ms bound
5	Delay sensitive appl., 100 ms bound
4	Delay sensitive appl., no bound
3	Currently unused
2	Currently unused
1	Less than Best Effort
0	Default service (Best Effort)

the IntServ model, however, is only coarse since individual flows of the same traffic class cannot be distinguished. If the high-priority traffic uses only a small fraction of the available transmission capacity, this may be sufficient to achieve a good approximation of the IntServ model. In this case, a very high percentage of frames is delayed at a bridge by at most the transmission time of a maximum sized frame. The percentage of frames experiencing longer queueing delays may be negligible for practical purposes. Admission control can limit the amount of time-critical traffic offered to bridges and thus influences the fraction of frames suffering longer queueing delays.

The semantic of the user priorities is defined in [2] as shown in Tab. 1. Delay sensitive applications that need quantifiable queueing delay bounds can use the user priorities 5 and 6, depending on their requirements. Following the IEEE 802.1D specification we consider the delay bound values given in Tab. 1 as the total maximum queueing time of a frame across an entire L2 domain.

2.1 Topologies of IEEE 802 LANs

L2 domains may have arbitrary topologies in which loops and multiple paths between edge devices may exist. Two different approaches exist to determine a path that a frame will follow across a 802 LAN:

- the IEEE 802 spanning tree protocol,
- source routing.

In the specification for IEEE 802 LAN bridges, a spanning tree protocol is defined that overlays the actual LAN topology with a spanning tree that contains exactly one path between each pair of LAN segments. Ports of bridges that are not part of the spanning tree are disabled. This establishes a virtual, connected, loop free topology, called the *active topology*.

If source routing is used, L3 devices that send frames onto L2 segments have to include into the frame header the exact path that the frame has to follow across the LAN to the destination L3 device. To learn the route to a given destination, a L3 device can send out *route explorer frames* that traverse

the network and reach the destination which sends back a response including the path taken by the explorer frame. Explorer frames can be forwarded by bridges using broadcasting (*All Routes Explorer frames*), allowing the discovery of all routes to the destination devices, or following a configured spanning tree (*Spanning Tree Explorer frames*) of the network, resulting in a unique route to the destination.

Ethernet devices use the spanning tree protocol while Token Ring and FDDI devices mainly implement source routing and use the spanning tree protocol only to a smaller extend.

3 Admission Control

Admission control must be employed offer the required QoS to data flows. Recently, queueing delay bound for networks with aggregate scheduling have been derived in the context of Differentiated Services networks offering Expedited Forwarding Service [1]. It can be shown that these queueing delay bounds are also applicable to L2 domains with priority queueing and multiple priorities.

In principle, these queueing delay bounds could be used to decide whether a new flow requiring a guaranteed maximum queueing delay can be accepted in a L2 domain or not. A flow would have to be refused if the delay bounds would exceed the allowed value. However, the service offered by an interface to the flows of a given user priority p depends on the traffic of all flows with priorities higher than p . Therefore, the admission of a high priority flow at an interface influences the service offered to lower priority flows, and consequently also the queueing delay bound that can be guaranteed to these flows. This renders necessary a complex admission control model which has to consider the possible changes of the queueing delay bounds provided to low priority flows in order to decide if a high priority flow can be accepted.

To reduce the complexity of the admission control decision we have developed an admission control algorithm that limits the traffic in each priority class such that the defined queueing delay bounds can be guaranteed. This admission control is based on the following assumptions:

- Each flows f subject to admission control is conform to a token bucket traffic controller with token rate r_f and bucket depth b_f upon entry to the L2 domain. This can be achieved by shaping the flows to their IntServ traffic specification.
- For each priority p there exist a constant τ_p , such that for all interfaces l of the L2 domain

$$\sum_{f \in S_{l,p}} b_f \leq \tau_p \sum_{f \in S_{l,p}} r_f, \quad (1)$$

where $S_{l,p}$ is the set of all flows with priority p that traverse interface l .

The parameters τ_p represent bounds on the sum of the bucket depths of the flows using a user priority p with respect to the sum of their mean rates. Often, for a given type of flows (e.g., voice or video flows) there is a linear relationship

between their bucket depth and their mean rate. However, τ_p also depends on the mix of flow types using a given user priority p . Therefore, we suppose that the actual values of the parameters τ_p will in most cases be determined empirically by the network administrator using measurements and used as a configuration parameter of the admission control algorithm. Under these assumptions, the maximum rate that can be admitted to each priority class with queueing delay bounds can directly be computed. The admission control algorithm therefore becomes very simple. It suffices to assure that the sum of the mean rate of all admitted flows is at most equal to the maximum admissible rate.

It turns out that the maximum admissible rate for a priority class at an interface depends on the ‘distance’ of the interface from the edge of the L2 domain. Formally, we define the *eccentricity* of an interface l as the maximum number of interfaces that any loop-free path between any interface of the L2 domain and the interface l , exclusive, may traverse¹. It follows from this definitions that the eccentricity of an ingress interface l to the L2 domain is 0, since there is no path to l from any interface different from l . We define the *diameter* of a network as the maximum number of interfaces that any loop-free path from an interface l to an egress node may traverse.

4 Comparison of Network Configurations

The maximal admissible rate for different traffic classes limits the transmission rate that can be made available at an interface for data flows requiring queueing delay bound guarantees. The admissible rate depends on factors like the network size and topology or the LAN technologies used. This must be considered in the planning of a network in order to correctly dimension the link capacities for the estimated traffic load.

4.1 Full Duplex Switched Ethernet Domains

As a first network configuration we consider a L2 domain employing switched Ethernet in the entire domain, as shown in Fig. 1. Depending on the capacity requirements, Ethernet (10 Mbits/s), Fast-Ethernet (100 Mbits/s) or Gigabit-Ethernet (1 Gbits/s) is installed in the different parts of the network. Although the end-systems have knowledge about individual flows, they are assumed to use aggregate scheduling on the interfaces transmitting into the L2 domain. Voice flows with an estimated value of $\tau = 0.025$ are mapped to the 10 ms delay bound service whereas video flows with $\tau = 0.09$ use the 100 ms delay bound service. The influence of network control traffic is neglected.

The delay bounds of the service classes are for the crossing of the entire L2 domain. The delay bound that has to be guaranteed at each interface is thus obtained by dividing the end-to-end delay bound by the diameter of the L2 domain, which has a value of 6 interfaces in this case.

¹ This definition is compatible with the definition of eccentricity in graph theory.

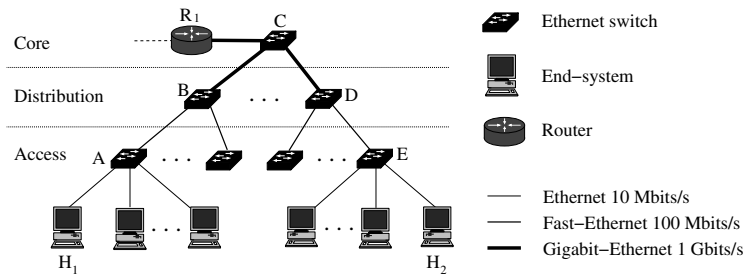


Fig. 1. Switched Ethernet L2 domain

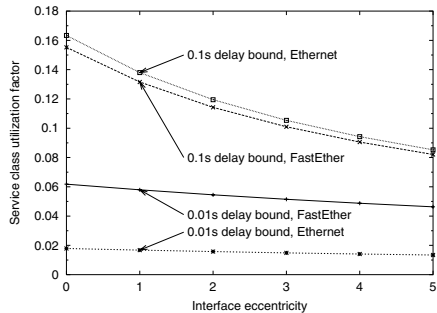


Fig. 2. Service class utilization factor of a switched Ethernet L2 domain

We want to evaluate the maximum share of the transmission capacity of each interface that can be used for the service classes with delay bounds. Therefore, we define a *service class utilization factor* as the ratio of the maximum admissible rate R of a service class and the interface transmission capacity C .

Fig. 2 shows the service class utilization factors of the considered configuration for different interface eccentricities. The values for Gigabit-Ethernet are not shown since they are very close to those of Fast-Ethernet. The figure has to be interpreted as follows. Consider the path from host H_1 to host H_2 . The interface $H_1 \rightarrow A$ has an eccentricity of 0 and uses Ethernet 10 Mbits/s, therefore a maximum of 1.8% of the interface capacity can be used for data flows requiring a 10 ms queueing delay bound. For the 100 ms queueing delay bound service class, at maximum 16.3% of the interface capacity can be used. The interface from A to B has eccentricity 1 and uses Fast-Ethernet, hence at maximum 5.8% and 13.2% of the interface capacity can be offered to the 10 ms and 100 ms queueing delay bound service classes, respectively. It can be seen that the service class utilization factors decrease with the eccentricity of an interface. In the extreme case of the interface with eccentricity 5 from switch E to host H_2 only 1.3% and 8.5% of the interface capacity are available for the two service classes.

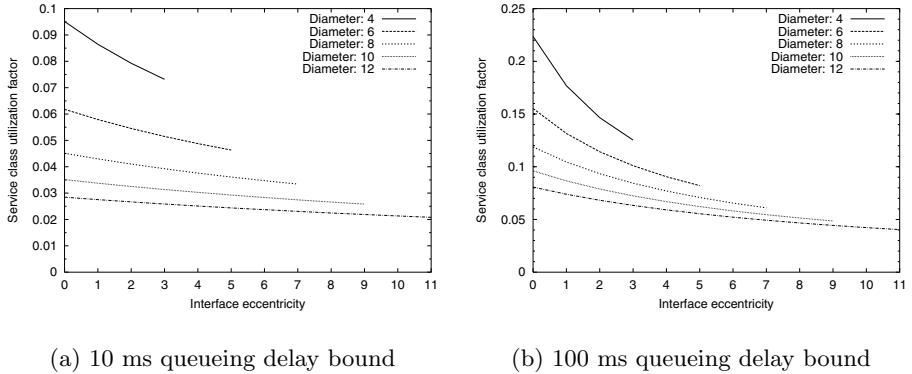


Fig. 3. Service class utilization factors in Fast-Ethernet network of different sizes

Influence of the Network Diameter

Increasing the diameter of a L2 domain decreases the service class utilization factors due to two effects. Firstly, the queueing delay bound at each individual interface decreases since more interfaces may be traversed by a flow and the end-to-end queueing delay bound across the entire L2 domain must not change. Secondly, interfaces may have higher eccentricities what reduces the maximum admissible rate. Fig. 3(a) and Fig. 3(b) show the maximum service class utilization factors at Fast-Ethernet interfaces for different interface eccentricities and network diameters. It can be seen that especially for the 10 ms queueing delay bound service class, the diameter of the L2 domain strongly influences the utilization factor. For example, at an interface of eccentricity 3, the utilization factor for the 10 ms service class changes from 7.3% to 3.9% as the diameter doubles from 4 to 8. For the 100 ms service class, 12.5% and 8.4% utilization can be achieved at an interface of eccentricity 3 for a network diameter of 4 and 8, respectively.

4.2 Token Ring Networks

The simplest configuration of a Token Ring LAN network consists of a single shared ring segment to which all devices of the LAN are connected, as shown in Fig. 4(a). A delay of $N \cdot \text{THT}_{\max}$ (including medium access delay and maximum frame time) may arise at an interface on a shared Token Ring segment before the highest priority queue can be served. Here, N is the number of stations sending high priority traffic and $\text{THT}_{\max}$ is the maximal token holding time. In general, we can assume that all connected stations may send high priority traffic. The default value for $\text{THT}_{\max}$ of 10 ms is too high to realize the 10 ms queueing delay bound service. To choose an appropriate value of $\text{THT}_{\max}$, the number of connected stations and also the desired utilization factors for low delay bounds services should be taken into account, since these services are primarily influenced by the medium access delay. Fig. 5 show the utilization factors on a

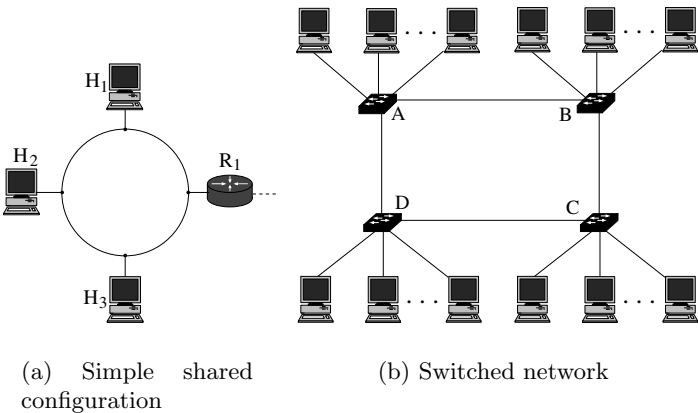


Fig. 4. Token Ring configurations (16 Mbits/s)

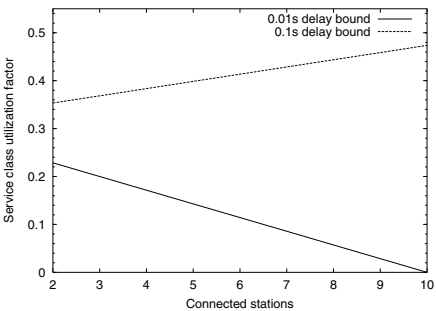


Fig. 5. Service class utilization factor of a single shared Token Ring segment

single 16 Mbits/s Token Ring segment with $THT_{max} = 1$ ms and a varied number of connected stations. Since this L2 domain has a diameter of 1, high utilization factors can be achieved for a small number of connected stations. However, the utilization factor for the 10 ms queueing delay bound service decreases linearly with the number of connected stations, due to the increasing influence of the medium access delay. For 10 or more connected stations, this service cannot be realized anymore, since the sum of the medium access delay and the maximum frame time is greater than the required queueing delay bound. The utilization factor for the lower priority 100 ms queueing delay bound service increases with number of connected stations up to 10 stations, because of the reduction of the higher priority traffic.

The Token Holding Time THT_{max} limits the maximum frame size that can be used by station. For $THT_{max} = 1$ ms, the maximum frame size is about

Table 2. Utilization factors for source routed and spanning tree Token Ring configurations

Interface Eccentricity	Source Routing		Spanning tree	
	10 ms service	100 ms service	10 ms service	100 ms service
0	6.0%	23.3%	4.0%	19.1%
1	5.5%	18.4%	3.7%	15.7%
2	5.0%	15.2%	3.4%	13.3%
3	4.6%	12.9%	3.2%	11.6%
4	—	—	3.0%	10.2%

2000 bytes. In order to avoid excessive protocol header overhead, its value cannot be arbitrarily small. Consequently, the number of stations that can be connected to a shared Token Ring segment is very limited. Separating the LAN into multiple shared segments interconnected by bridges does not improve the situation since this leads to longer paths across the entire LAN and therefore smaller per-interface queueing delay bounds. We can therefore conclude that shared Token Ring segments are not well suited to implement services with low queueing delay bound guarantees. The solution to this is micro-segmentation, i.e., the use of half-duplex or full-duplex switched Token Ring segments with only two respectively one sender per segment. Nevertheless, a small maximum frame size should be chosen also in these cases to avoid long delays for high priority packets due to the transmission of big lower priority packets. In the following we will therefore again assume a maximum Token Holding Time of 1 ms, which also determines the maximum frame time.

Token Ring networks by default employ general source routing which allows discovering the best paths between two nodes of the network. This has the advantage, that the traffic load on the segments can be balanced. Fig. 4(b) shows a switched Token Ring network consisting of four interconnected 16 Mbits/s full-duplex Token Ring switches. If general source routing is employed and assuming that the end-systems always choose the shortest path to transmit frames, at most four transmission interfaces may be traversed by any flow inside this L2 domain. The transmission interfaces at the end-systems have eccentricity 0. The interfaces of the switches towards the end-systems have the maximum eccentricity of 3. All other interfaces have a eccentricity of 2.

Source routing Fig. 4(b) may also be based on a spanning tree configuration. Therefore, switch A may be chosen the root of the spanning tree and the segment between C and D may be blocked. This configuration has a maximum path length of 5 and a maximum interface eccentricity of 4. Tab. 2 compares the resulting service class utilization factors of the two configurations. Especially in the case of the 10 ms queueing delay bound service, the achievable utilization factors for equal interface eccentricities are substantially lower for the spanning tree configuration. This is due to the increased number of interfaces to traverse in this configuration. The main difference between the two configurations can be found at the interface $A \rightarrow D$. This interface has eccentricity 2 in the source-

routed configuration and eccentricity 3 in the spanning tree configuration. This means that 5.0% of the transmission capacity are available for the 10 ms queueing delay bound service in the former configuration compared to only 3.2% in the later. For the 100 ms service, its utilization factor decreases from 15.2% to 11.6% when using the spanning tree protocol. Also, the traffic experienced on this interface may be higher in this case, since all flows from *A*, *B*, or *C* towards *D* have to traverse this interface. In the source-routed configuration, only flows from *A* to *D* and about half of the flows from *A* to *C* and from *B* to *D* use this interface. We thus find that the spanning tree topology introduces a potential bottleneck at the root of the tree that does not exist in the source-routing configuration.

5 Conclusions

In this article, the possibilities to realize the IntServ model in IEEE 802 LANs as well as the limiting factors have been presented. The first part gives a description of the mechanisms defined by the IEEE that make it possible to support Integrated Services in such networks. We have seen that the major difference of layer 2 devices compared to layer 3 devices is the impossibility to identify individual data flows on the former.

For the dimensioning of a LAN it must be known which traffic intensity may be admitted on the network interfaces for different service classes. Therefore, it is necessary to define the admission control algorithm used on the interfaces. An admission control algorithm based on the queueing delay bounds result would be very complex and thus is unlikely to be implemented on layer 2 devices. We have therefore developed a simplified admission control algorithm that allows us to explicitly compute the maximum mean rate of the aggregate flows of the different service classes that can be accepted at an interface of a given transmission capacity.

We use the developed model to compare LAN configurations with respect to the utilization factors that can be achieved at interfaces for different service classes. The main result is that for configurations of realistic size in general only a small fraction of the total interface capacity can be used for flows requiring queueing delay bounds. We show how the achievable utilization factors are influenced by the network size, the LAN technologies used and the routing scheme of frames across the LAN.

References

1. A. Charny and J. Y. Le Boudec. Delay bounds in a network with aggregate scheduling. Technical Report DSC/2000/022, EPFL-DI-ICA, April 2000.
2. M. Seaman, A. Smith, E. Crawley, and J. Wroclawski. RFC 2815: Integrated service mappings on IEEE 802 networks, May 2000. Status: Proposed standard.

Integrating Differentiated Services with ATM^{*}

Sowmya Manjanatha and Radim Bartoš

Dept. of Computer Science, University of New Hampshire, Durham, NH 03824, USA,
{sowmyam,rbartos}@cs.unh.edu, Phone: (603) 862-3792, Fax: (603) 862-3493,
<http://www.cs.unh.edu/~rbartos>

Abstract. IP in the edge and ATM in the core are commonplace in today's internetworks. The IETF has proposed a new Quality of Service (QoS) mechanism namely Differentiated Services (DiffServ) for IP networks. On the other hand, QoS is an inherent feature in ATM. It is imperative that IP and ATM QoS interoperate efficiently to provide an end-to-end service guarantee. DiffServ provides a class of service named Assured Forwarding (AF) that does not exactly correlate to any of the service categories offered by ATM. AF is targeted towards a range of applications, such as real-time (rt) that do not require a constant bit rate service provided by Expedited Forwarding, and other non-real-time (nrt) applications that expect a service better than Best Effort.

In this paper we propose the mapping of AF to the Variable Bit Rate (VBR) service category in ATM. VBR is suitable because it is available in the form of rt-VBR and nrt-VBR and could be translated appropriately based on the applications. The mapping is implemented and verified using the LBNL Network Simulator. The results of the experiments show that VBR is a better match for AF than any other service category in ATM.

1 Introduction

Recent advances in communications has facilitated computer networks to support a wide spectrum of applications such as voice, multimedia, and traditional data. The introduction of voice and multimedia demands stringent service requirements such as bounded end-to-end delay, and delay variance in addition to a guaranteed traffic delivery mechanism. Quality of Service is envisioned as an essential component in building efficient networks.

Several efforts in the area of QoS has resulted in approaches such as Integrated Services (IntServ) [1], MPLS traffic engineering [2], and Differentiated Services [3] in the IP domain and ATM Traffic Management Specification [4] in the ATM domain. IntServ offers an end-to-end service guarantee with Resource Reservation Protocol (RSVP) [5] as the signaling tool to reserve resources at every node in a path for every flow. The reservations are maintained in these nodes using a soft-state database imposing a very high demand for processing time and state maintenance storage in the backbone routers. MPLS Traffic Engineering is an ongoing effort by the Internet Engineering Task Force (IETF).

^{*} This research was supported in part by NSF-ARI grant No. 9601602.

A different QoS approach, Differentiated Services (DiffServ), has been recently proposed by the IETF. DiffServ attempts to reduce the processing time by pushing the functional elements required to implement QoS towards the edges of a network. QoS provisioning is based on aggregates of flows that further reduces the state information maintained on individual routers.

There are advantages to both IP and ATM technologies which have necessitated their co-existence in the network infrastructure. For instance, the ability of IP to adapt rapidly to changing networking conditions makes it appropriate for core routers. On the other hand, the scalability and cost/performance model of ATM switches are appropriate for backbone networks. The interoperation of the features of the two technologies to provide end-to-end QoS is crucial for the emergence of fast and reliable next-generation networks.

One of the issues in integrating IP DiffServ with ATM QoS is translating the Assured Forwarding Per Hop Behavior (AF PHB) [6] service requirements on to the ATM domain. The AF PHB is targeted towards a range of applications whose service requirements may vary from a level better than best-effort to applications that require a minimum guaranteed rate and delay characteristics. Additionally, AF introduces a concept of relativity that allows multiple AF aggregates of a class to be provisioned relative to one another.

In this paper we propose to map the AF PHB to the VBR service category of ATM. VBR is attractive because of its ability to serve both real-time and non-real-time applications. Through simulation experiments, we show that AF relativity concept can be achieved by tuning the traffic parameters of different VBR connections mapped to a single class.

2 Background

The task of integrating IP Differentiated Services and ATM QoS is not straightforward because of their inherent implementation differences. One of the major difficulties in merging the QoS architecture of the two technologies is that there is no service category in ATM that is similar to that of AF PHB. AF was developed to support those applications that required a minimum guaranteed rate or end-to-end delay but did not need a channel dedicated to them such as in the Premium Services. Additionally, AF incorporates the concept of relativity whereby customers have the ability to prioritize different flows emerging out of their domain. Although, AF has many attractive features, its deployment will be difficult if there were no efficient mechanisms to integrate it with other technologies.

The problem of mapping AF to an appropriate ATM Service category has caught the attention of several researchers. Rabbat et al. have proposed two mapping mechanisms to ABR [7] and GRF [8]. In both the schemes, the focus was on the effective throughput and AF relativity. Rogers et al. [9] suggested a mapping of AF to VBR in their study of a new shaping algorithm for DiffServ. The scope of their study [9] was the traffic conditioning mechanisms for Differentiated Services. The study of the performance characteristics, end-to-end delay,

and jitter in particular for mapping the real-time categories of AF to ATM is yet another interesting research topic.

This paper develops a framework to map the AF PHB to the VBR service category in ATM. We manipulate the advantage in VBR to match all the types of applications targeted by AF. We further show that relativity can be achieved by tuning the traffic parameters used for different VBR services. The proposed architecture is verified using simulations using the LBNL (Lawrence Berkeley National Laboratory) Network Simulator (*ns*) [10].

3 Proposed Mapping

In designing the architecture, there are two issues to be considered: (i) the position of the mapper in an intermixed IP and ATM network, (ii) the QoS parameters that must be mapped.

The translation of DiffServ to ATM must happen at the IP boundary on a per-aggregate basis. Translation in the ATM domain may lead to complications due to the connection oriented nature of ATM. Each AF aggregate exiting a DiffServ domain would be mapped to a different VBR Virtual Circuit (VC) in an ATM domain. The real-time aggregates (for example, multimedia applications) are mapped to rt-VBR (real time VBR) and non-real-time applications are mapped to nrt-VBR (non-real time VBR) [4]. In ATM, the traffic parameters corresponding to a service category is accepted at connection establishment time through the Connection Admission Control (CAC) procedure.

In case of VBR, the parameters that constitute the service characteristics are the Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), Maximum Burst Size (MBS in cells), Cell Delay Variation Tolerance (CDVT). The service parameters used in AF are Peak Information Rate (PIR), Committed Information Rate (CIR), Maximum Burst Size (MBS in packets) and Packet Delay Variation. Packet Delay Variation is an optional parameter and is mostly used when the application is real-time. The mapping from AF to VBR is done as follows:

- PIR to PCR.
- CIR to SCR.
- PDV/cells per packet to CDVT
- MBS*packetsize to MBS*cellsize

It is important to tune SCR and CDVT for the real-time applications. In case of the AF relativity feature, the relative priority is usually assigned on the basis of the amount of bandwidth shared at a particular time in transmission. Therefore, the important parameters to consider are the SCR and the MBS. Other parameter to consider is the Cell Loss Priority (CLP). The CLP is particularly useful after connection establishment. All the packets that conforms to SCR are marked as *good* (CLP=0) and the non-conforming ones are marked as *bad* (CLP=1). In case of DiffServ, there are three levels of drop precedence while CLP can be assigned only two values. To address this issue, packets arriving at a rate

$$Mapped\_SCR = SCR + \delta_{SCR},$$

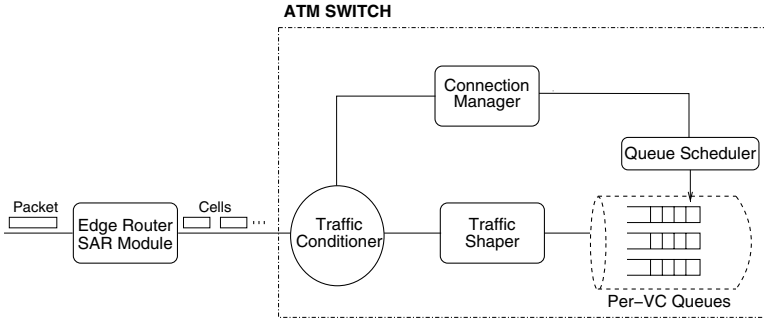


Fig. 1. Architecture of ATM implementation on *ns*.

where $\delta_{SCR} = \pm 10\%$ of SCR value are marked as good. The other option is to use the VBR3 [4] category of ATM in which, cells are tagged and service degraded instead of cells being discarded during times of congestion.

4 Simulation Setup and Experiments

The LBNL Network Simulator (*ns*) with the DiffServ and ATM enhancements was used for our experiments. The Simulator has the facility to simulate IP networks with the RSVP and DiffServ QoS mechanisms. We enhanced the Simulator to incorporate ATM functionality as well.

4.1 ATM Simulator

The ATM feature added included two main components, an *ATM End Station* and an *ATM Switch*. The ATM Switch consists of a *Connection Manager*, *Traffic Conditioner* and a *Queue Scheduler*. Figure 1 depicts the design of the ATM Simulator.

The *Connection Manager* provides the functions to create, and delete ATM Permanent Virtual Circuits (PVC), and lookup the created PVC Database. The *Traffic Conditioner* performs the Connection Admission Control (CAC), and the Traffic Policing/Usage Parameter Control (UPC) and the Traffic Shaping functions. The *Queue Scheduler* schedules the traffic on the link. Queuing is done on a per-VC basis to provide fairness to all traffic especially during congestion. Two different scheduling mechanisms namely Priority and Weighted Round Robin (WRR) were considered. On the high level, priority is given on the basis of ATM QoS classes, i.e., 0 for CBR, 1 and 2 respectively for rt-VBR and nrt-VBR, 3 for ABR, 4 for GFR and 6 for UBR. Between the various VC Queues of each category, Weighted Round Robin scheduling was used. The weights depend on the Peak Cell Rate (PCR) for CBR, Sustainable Cell Rate (SCR) for real time and non-real time VBR, Minimum Cell Rate (MCR) for ABR and GFR. For UBR the weights assigned to all VCs were same since the category is best-effort.

The *ATM End Station* provides the facility to the perform the segmentation of IP packets to cells and reassemble cells to IP packets using the ATM Adaptation Layer 5 (AAL5) protocol.

4.2 Topology and Experiments

A network topology used by most researchers for the study of QoS is shown in Figure 2 [8]. There are 12 sources (S1 ... S12) and destinations (D1 ... D12) on either side of a core network consisting of 6 Edge Routers (ER1 ... ER6) and two ATM switches (SW1 and SW2) separated by a bottleneck link as shown. All the links from the sources to the Edge Routers and Edge Routers to destinations were 6 Mbps. The links from Edge Routers to Switches and vice versa were 25 Mbps. The bottleneck link was 40 Mbps. The links were chosen such that the only bottleneck in the network was the core, i.e., the link between switches. A small propagation delay was also accounted for and it was a value of 5ms for all the links. The traffic sources used were CBR with UDP Transport Agent as real time generators and FTP with TCP Transport agents as non-real time generators. At the sources, each traffic flow is assigned to one of the four different AF classes (Platinum, Gold, Silver, and Bronze). The relatively low transmission rates were chosen in order to keep the number of packets generated and hence the simulation times at a reasonable level.

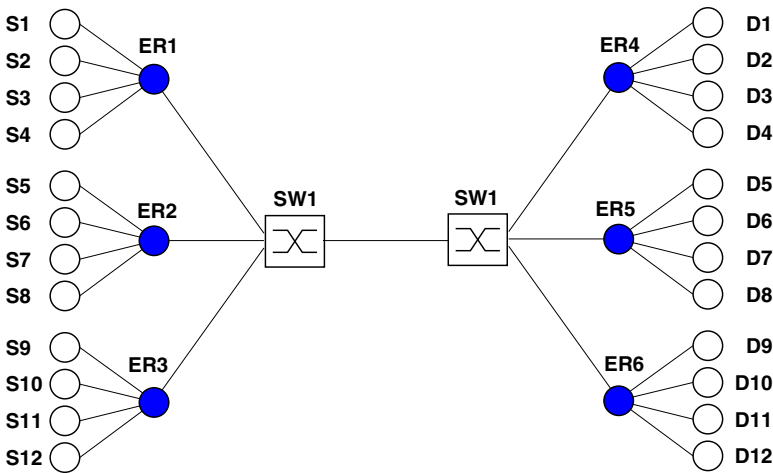


Fig. 2. Network topology for experiments.

Two sets of experiments, (i) to test the performance characteristics (throughput, end-to-end delay, and jitter) of real-time sources with non-real-time sources, (ii) to test AF relativity were conducted. The experiments involved varying the source rates, the queue lengths and tuning the parameters, i.e., SCR, PCR and

the MBS. Traffic entering the Edge Routers (ERs) are scheduled with differentiation performed using DiffServ. At the edge the segmentation function is applied to convert packets to cells before scheduling them on the link.

For the first set of experiments, three different experiments were conducted. In the first experiment, performance measurements of the network without any ATM, i.e., with two core DiffServ enabled IP routers were obtained. For Experiments 2 and 3, three PVCs were added one between each incoming and outgoing Edge Router. The aggregated traffic from 4 sources on each edge was transmitted on a single PVC. Each of the PVC was associated with a Traffic Descriptor that includes PCR, SCR, Maximum Burst Size (MBS) and Cell Delay Variation Tolerance (CDVT) for real-time and non-real-time VBR. The traffic parameters were assigned according to the mapping explained in Section 3. For the second experiment, we obtained results by mapping DiffServ to UBR service category. For the third experiment, we had DiffServ mapped to VBR. As explained in Section 3, we mapped traffic parameters of ER1 to rt-VBR (since this received traffic from CBR sources) and traffic parameters of ER2 and ER3 to nrt-VBR but the parameter values were different for ER2 and ER3. The first set of experiments was as follows:

Experiment 1: The source rates of CBR sources were varied keeping the Committed Information Rate (CIR) value in Experiment 1 and the equivalent mapped SCR in Experiment 3 constant. The variance of throughput, delay and jitter in Experiments 1, 2, and 3 were studied. The variance of transmission rates of sources attached to TCP agents are not necessary since the TCP sources adjust their rates according to the feedback from the network.

Experiment 2: A study of how delay and jitter in the mapping of DS to UBR vary with queue lengths in the network was conducted.

For the second set of experiments, 6 PVCs were added, one between ER1, ER4 pair, one between ER3, ER6 pair and 4 between ER2, ER5 pair. In this experiment, the traffic parameters used on ER1 and ER4 were pertaining to the EF service category of DiffServ and they were mapped to the CBR service category in ATM. The ER2, ER5 pair were configured to perform service differentiation using 4 different AF codepoints to yield AF relativity. The 4 PVCs between ER2 and ER5 correspond to 4 different codepoints used on ER2 and ER5. All the 4 PVCs were associated with nrt-VBR service category but with SCR equal to the Committed Information Rate (CIR) associated with each codepoint on the edge routers. The PVC between ER3 and ER6 was associated with UBR service category. In this study, the following experiment was performed:

Experiment 3: The source rates were kept constant, and the SCRs mapped to different CIRs for the 4 different codepoints of the PVCs mapped correspondingly from the ER2 were varied. For each of the variations, the SCRs of PVC3 was 75% of the SCR for PVC2, SCR of PVC4 was 50% of PVC2 and SCR of PVC5 was 25% of PVC2. The throughput for the 4 PVCs were verified to be relative to each other.

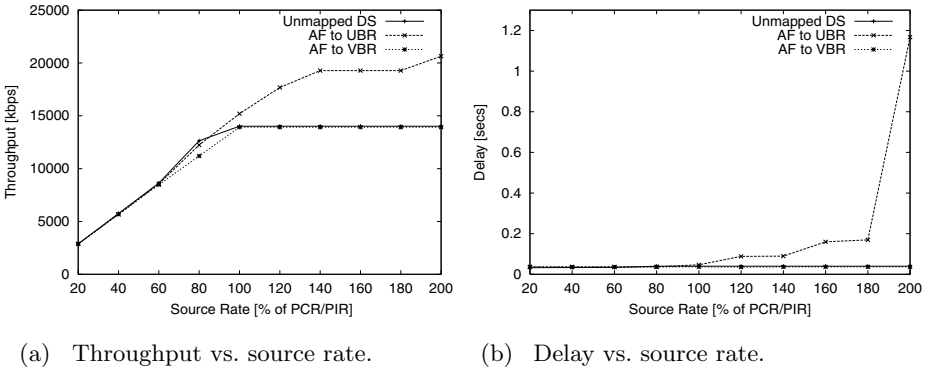


Fig. 3. Results of Experiment 1.

5 Simulation Results

For the first set of experiments explained in Section 4, the importance is laid on the behavior of real-time applications in AF mapped to the rt-VBR category. Therefore, the following results pertain to the total achieved rate, average delay, and average jitter obtained from the sources S1 through S4 which are aggregated to a single code point on router ER1.

Figures 3(a), 3(b), and 4(a) display the results of Experiment 1. In each of the graphs, the behavior of the network without ATM and with ATM were studied. The expected performance of the network here is that guaranteed service be provided if the source behaves as requested. Since, the source lays a stringent requirement on service, non-adherence to service must be treated strictly by policing out excess traffic. The graphs clearly display that UBR neither maintains the consistency in delay and jitter nor does it police traffic entering beyond the requested rate. The network consists of TCP and CBR sources. Allowing excess traffic for the CBR sources causes recession in bandwidth. TCP sources depend on the feedback obtained from the network and therefore excess CBR allowed rate causes a reduction in TCP achieved rate. The achieved rate of TCP sources varied between 3 and 15 Mbps when the bandwidth available after all the CBR sources could be accommodated was 20 Mbps in the case when the total source rate of CBR sources was 20 Mbps. The behavior of UBR category is undesirable as this leads to starvation of low priority sources in the network. In case of DS and VBR, we saw that the achieved TCP rates were approximately 25 Mbps because the traffic was policed at 15.8 Mbps (PIR/PCR). We only see about 14.4 Mbps sustained rate because the CIR/SCR agreed was 14.4 Mbps.

Figures 4(b) and 5(a) present the results of Experiment 2. The UBR/ABR and the GFR scheduling mechanism used in the ATM switches is designed to utilize the bandwidth in the network to the fullest possible extent. The queue sizes in the switches are dynamically allocated until the maximum threshold of the system is reached. The sizes are allocated with respect to the incoming

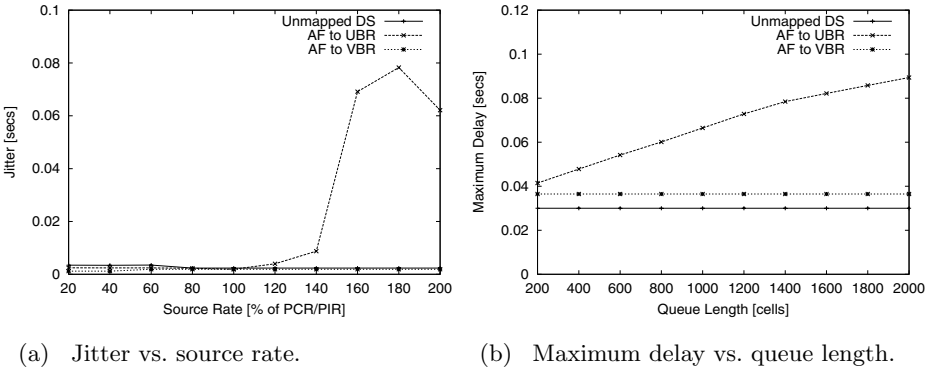


Fig. 4. Results of Experiments 1 and 2.

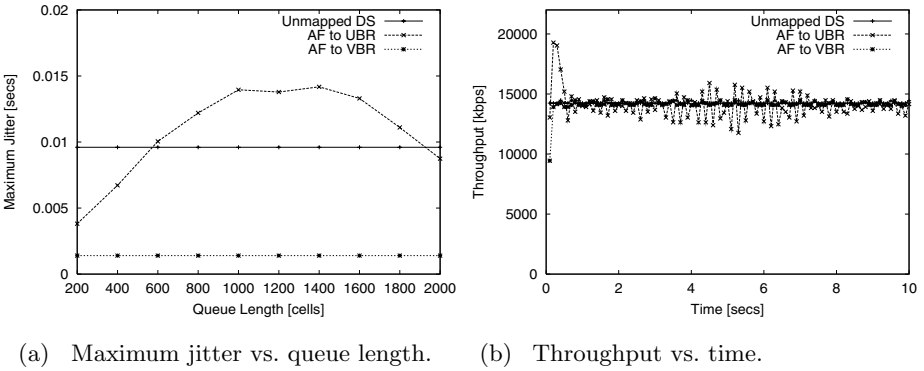
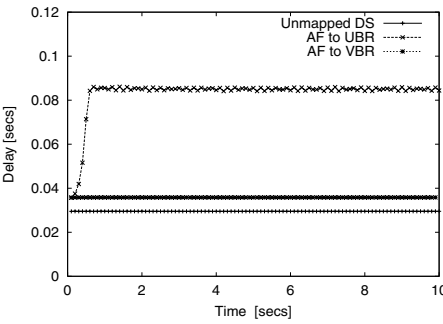


Fig. 5. Results of Experiment 2.

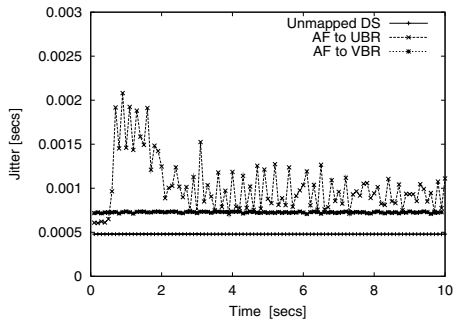
traffic. The delay and jitter experienced by these service categories is mainly due to the scheduling. In these categories, since traffic is not strictly policed beyond the guaranteed rates, traffic is not dropped until the maximum size is hit. In case of VBR and AF (higher codepoints only), the buffer sizes do not affect the delay and jitter due to strict policing. The figures display this behavior. This experiment also shows that congestion in the network and thereby multiple retransmissions of data, that further aggravates the condition of the network, can be avoided when a proper check is put to malicious resource utilization.

Figures 5(b), 6(a), and 6(b) present the additional results of Experiment 2. A source rate of 5 Mbps per source is picked for the study. The real-time applications expect that the performance remain constant with time. The graphs indicate the consistency.

In the second set of experiments explained in Section 4, the relativity is basically the measure of relative throughput to be maintained through the AF PHB group. In this experiment, the AF PHB group constitute sources S5 through S8.



(a) Delay vs. time.



(b) Jitter vs. time.

Fig. 6. Results of Experiment 2.

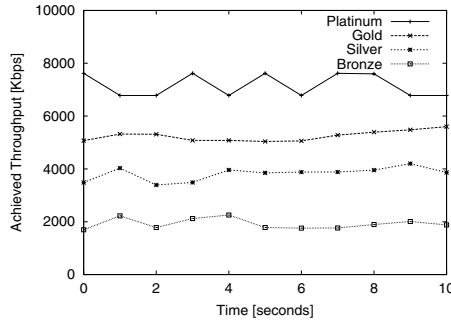


Fig. 7. Experiment 3: AF relativity with VBR mapping.

Figure 7 presents the results of Experiment 3. Comparing the results to that of the experiments by Rabbat et al. [8,7], we see that the relativity obtained is similar. We verified the relativity feature with different sets of CIR to SCR values. Table 5 displays the results. The first column in the table contains the CIR/SCR used for the Platinum source. The CIR/SCR values of the gold, silver and bronze were each 25% less than the value of the next higher level as explained in Section 4.

6 Conclusion

In this paper, the importance of QoS with an emphasis on efficient interoperation of QoS in IP and ATM networks was discussed. A framework for the translation of AF PHB to the VBR service category was put forth. The experimental setup and the results show that VBR is a suitable category for all types of applications targeted by the AF PHB. It was further shown that AF relativity can be achieved by tuning the VBR traffic parameters.

CIR/SCR of Platinum (Mbps)	Achieved Throughput			
	Platinum	Gold	Silver	Bronze
5.088	5.0	3.8	2.54	1.2
4.24	4.16	3.18	2.12	1.06
3.392	3.4	2.52	1.56	0.81
2.544	2.5	1.9	1.26	0.6

Table 1. Achieved rates of the mapped AF Olympic classes

References

1. J. Wroclawski et al., "The use of RSVP with IETF Integrated Services," IETF RFC 2210, September 1997.
2. D. Awduche et al., "Requirements for Traffic Engineering over MPLS," IETF RFC 2702, September 1999.
3. S. Blake et al., "An Architecture for Differentiated Services," IETF RFC 2475, December 1998.
4. Technical Committee, "Traffic Management Specification - version 4.1," ATM Forum approved specification AF-TM-0121.00, March 1999.
5. R. Braden et al., "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification," IETF RFC 2205, September 1997.
6. J. Heinanen et al., "Assured Forwarding PHB group," IETF RFC 2597, June 1999.
7. R. Rabbat et al., "Supporting Differentiated Services using ATM ABR Service," International Conference on Computer and Communication Networks, ICCCN'99, Boston, MA, October 1999.
8. R. Rabbat et al., "Differentiated Services Support using ATM GFR Service," Parallel and Distributed Computing and Systems, pp. 455-460, Las Vegas, NE, November 2000.
9. G. Rogers et al., "Matching Differentiated Services PHBs to ATM Service Categories - the AF to VBR Case," IEEE LANMAN Workshop, September 2000.
10. S. McCanne and S. Floyd, ns - Network Simulator, <http://www.isi.edu/nsnam/ns>.

Management and Realization of SLA for Providing Network QoS

Manzoor Hashmani¹, Mikio Yoshida¹, Takeshi Ikenaga², and Yuji Oie²

¹ NS Solutions Corporation, Japan

7-20-1, Fukushima Osaka 553-0003, Japan

{hashmani.manzoor, yoshida.mikio}@osk.ns-sol.co.jp

² Dept. of Computer Science and Electronics, Kyushu Institute of Technology,

Iizuka-shi 820-8502, Japan

{ike, oie}@cse.kyutech.ac.jp

Abstract. We consider an SLA (Service Level Agreement) committed between two parties to use the guarantee of QoS provided by a QoS Enabled Network (QEN). QEN can provide guarantee of QoS because a Bandwidth Broker (BB) manages its resources based on a policy. The IETF and the DMTF have done extensive work in this field and have proposed a policy framework in order to store and manage policy and the standardization process is still going on. However, not much work has been done in the field of *realization* and *management* of SLA, which is necessary in order to utilize QoS provided by the network layer as perceived by the service layer. In this paper, we propose a new methodology of SLO templates (machine readable description of human readable textual SLA) to negotiate and commit SLA to utilize QoS provided by networks as perceived by businesses like that of a virtual leased line service. We also describe the implementation of our proposed SLO templates in our BB.

1 Introduction

The guarantee of quality of service (QoS) is one of the most important issues for businesses and common users to use the Internet for their mission critical applications [6]. The research community and standards organizations like the IETF (Internet Engineering Task Force) and the DMTF (Distributed Management Task Force) have been working to achieve this objective. In this concern, the IETF has proposed mainly two architectures called IntServ [4] and DiffServ [1] to provide guarantee of network level QoS. These architectures provide service guarantees to users by making network components (routers) differentiate traffic on a basis of policy, for differentiation between high priority and low priority traffic. For the maintenance of policy, the IETF and DMTF have proposed a framework called “Policy Framework” [9].

Part of the policy provided by the network administrator is derived from the SLA (Service Level Agreement) negotiated and committed between network service provider(s) (generally known as ISPs) and service consumer(s) (either ISPs or common users or both). DiffServ framework suggests that the negotiated and committed SLA in human readable textual form may be represented by SLOs (Service Level Objectives) for machine readability [13]. These SLOs consist of some

parameters and their values. Though this basic framework of SLOs has been defined but not much work has been done to use the parameters suggested by the DiffServ framework to create meaningful services as perceived by the users or applications.

We propose a new concept of SLO templates. We suggest ways to use these templates to create services useful for users or applications. The SLO templates are used to negotiate and commit SLAs, which are then enforced over DiffServ architecture by our bandwidth broker. These templates are designed in order to satisfy QoS requirements perceived at the service/application level and are implementable over DiffServ architecture. We expect that with time and usage some of the templates will be discarded, a few will become useful in local environments and the others may become standard services. In this paper, we also describe implementation of our proposed SLO templates using/in our BB.

The rest of the paper is organized as follows. Section 2, gives the current status of the research in the field of SLA and QoS in terms of DiffServ architecture. In this section, we also describe the motivation behind our proposal of SLO templates. Section 3, describes our proposed SLO templates. We describe the implementation of our proposed SLO templates in our BB in section 4 and also describe experiments performed. In section 5, we explain the lessons learnt during the implementation and experiments. We finally provide summary and conclusion in section 6.

2 Current Status of SLA and QoS

In this section, we briefly describe the current status of SLA and QoS strictly in terms of DiffServ architecture. This is because we want to focus on providing services over DiffServ architecture, which provides assurance of QoS and is supposed to be scalable at the same time. We neither intend to nor is it possible due to space limitation to cover these two fields in a broad sense.

2.1 Our Motivation: SLA for Network QoS

An SLA describes a high-level business policy and is converted into SLOs (Service Level Objectives) for machine readability. An SLO consists of parameters and their values. In DiffServ, SLO parameters are identifiable [14][15]. It is suggested that these parameters can be used to establish services like a Virtual Leased Line (VLL). The types of parameters contained in an SLO and their values may vary even to construct the same type of service depending on the negotiators of the SLO. On the basis of negotiators, we divide SLO into two types.

User-ISP SLO: It is committed between a common user and an ISP.

ISP-ISP SLO: It is committed and agreed upon between ISPs. We call this as Inter-Domain SLO because it is negotiated between ISPs on a per-domain basis.

The above stated have to be designed separately because the service requirements may be different in both cases. In this paper we focus on ISP-ISP SLO only.

At present, it is necessary to decide values of all of the parameters to construct a service. For a committed SLO, the values of these parameters are decided after negotiation between service providers and service consumers. Negotiating the values of these parameters is a very complex issue, which depends on many factors including but not limited to business model, service to be provided, technical limitations, etc. A concrete framework or study in this regard is not available. In this paper, we attempt to provide solution to this problem (see Sect. 3).

2.2 Policy Framework

One way of providing the service agreed upon in an SLO is to derive a set of policy rules and manage a network according to these policy rules. The policy rules do not necessarily consist of only the ones derived from SLO, but these may also be provided by many sources like network administrator etc. These policy rules can be stored and managed (translation, conversion, conflict detection etc) using a framework called Policy Framework (PF) proposed by the IETF. PF states that one global set of policy rules is neither flexible enough nor suitable to manage various policy rules. Therefore, it suggests management of policy rules at various levels of abstraction, namely, High-Level Business Policy, Device Independent Policy, and Device Dependent Policy (see Fig. 1). It might be necessary to convert high-level policy rules derived from SLO into device independent policy rules before being converted into device dependent policy rules which are finally enforced over the network components.

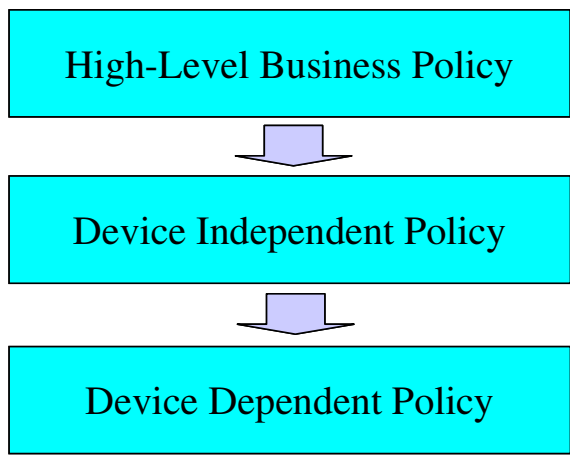


Fig. 1. Policy Framework showing the concept of managing policy rules into layers

2.3 Our BB

In 1999, an initiative was taken by CKP/NGI to provide guarantee of end-to-end QoS over DiffServ domain for contents business [6]. We developed a server to record and manage usage of network resources (bandwidth). We called this Bandwidth Broker (BB) as ENICOM's BB. Our BB performs admission control and router configuration on the basis of provided policy (Fig. 2). Since its first appearance, our BB has been constantly enhanced to include many new features and accommodate new standards as well [7][8]. We have performed several QoS experiments deploying our BB over wide area networks (WAN) in Japan. Though initially we did not design our BB to perform SLA negotiation, but due to its inherent feature of performing resource management and admission control on the basis of policy rules, it can be easily enhanced to negotiate and enforce SLA.

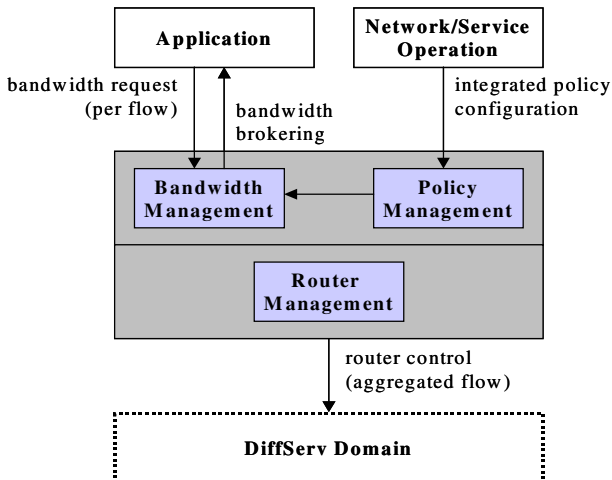


Fig. 2. Basic concept of our BB

3 Our Proposal

As we know, sufficient research has already been done to obtain guarantee of QoS from the network. In this regard, two models/architectures have wide popularity, i.e., Integrated Services (IntServ) and Differentiated Services (DiffServ). We have chosen DiffServ because it is more likely to scale well.

A well-known example of creating services over DiffServ is a service similar to a leased line (also called Virtual Leased Line or VLL). Due to its QoS capabilities, these services are supposed to be used by the critical business applications like contents business, Voice over IP (VoIP), video conferencing, telemedicine etc. But these can only be realized when service providers and consumers establish a service level agreement (SLA). Generally speaking, business SLAs are a combination of text and a set of parameters and their values. To make an SLA as machine readable, it is

represented by one or more Service Level Objectives (SLOs). An SLO is a set of parameters and their values. Fortunately, in case of DiffServ, some primitive parameters have already been identified [14][15]. But these alone are not sufficient and require identification of more parameters to finalize SLA.

In the next step, the service consumers and service providers need to negotiate values of these parameters. This process of negotiation is very cumbersome due to many reasons. For example, a large number of parameters make it difficult for service providers and consumers to find a suitable combination, which satisfies their needs. Some values of parameters may not be supportable (technically or otherwise) by service providers. This means renegotiation may be required.

3.1 SLO Templates

In order, to overcome the above stated problem, we propose a new idea of SLO templates. Each SLO template consists of variable and constant parameters (Table 1). Variable parameters are the ones whose values are negotiated between service providers and consumers. The constant parameters are those whose values are fixed and are not negotiable due to reasons like poor feasibility, resources-not- available, technical faults etc. For example, resources-not-available may limit the values of delay that can be supported.

In Table 1, we have listed those parameters that we use to create SLO templates based on the concept described in the above paragraph. It is not our intention to create an exhaustive list of parameters. Rather, our purpose is to list those parameters, which can be used to create SLO templates, which in turn may be used to create useful service for business applications. The first four parameters namely, PHB (Per Hop Behavior), BW (Bandwidth), BT (Burst), and DY (Delay) are directly related to DiffServ parameters and the agreed upon values of these parameters can be fulfilled using DiffServ parameters. On the other hand, the last two parameters namely, AY (Availability) and Cost Factor (CF) are high level parameters and the agreed upon values of these parameters can not be fulfilled using low-level parameters. Rather these are fulfilled using admission control in BB.

Table 1. Proposed SLO templates (the parameters whose values are not specified (given as xx) are variable parameters and the others are constant parameters)

Parameters	PHB	BW (Kbps)	BT (KB)	DY (msec)	AY (%)	CF
SLO Templates						
<i>Premium1</i>	EF	xx	xx	20	100	F0
<i>Premium2</i>	EF	xx	xx	30	90	F1
<i>Premium3</i>	EF	xx	xx	40	80	F2
<i>Premium4</i>	EF	xx	xx	50	50	F3

We now briefly explain AY and CF. The AY parameter indicates the maximum share of a sub-service (e.g. Premium1) in total amount of resources allocated to a service (e.g. Premium). For example, a 90% value of AY for Premium2 in Table 1

indicates that at maximum Premium1 will be allocated 90% of Premium resources. However, a comprehensive algorithm is yet to be developed to ensure strict compliance to the allocated share of resources. On the other hand, parameter CF indicates a factor, which is multiplied with the unit cost of the resource. For example, if the unit cost of 1 Kbps of bandwidth with a burst size of 100 KB is C, then for a Premium1 user availing only one unit, the total cost is calculated as $C \times F_0$.

3.1.1 Constant Parameters

The constant parameters are those whose values are not negotiable for any SLO template. For example, in Table 1 all four SLO templates (Premium1, Premium2, Premium3, and Premium 4) are designed to work only over Expedited Forwarding (EF) PHB of DiffServ. The other constant parameters are DY, AY and CF. These SLO templates are designed in such a way that **Premium1** is better than **Premium2**, **Premium2** is better than **Premium3** and so on. For example, delay for **Premium1** is less than **Premium2**. On the other hand availability and the cost factor are greater for Premium1 than Premium2. For example, $F_0 > F_1 > F_2 > F_3$.

3.1.2 Variable Parameters

In the table placing 'xx' in their column indicates these parameters. The values of these parameters are not predetermined and are decided after negotiation. We have decided to use Bandwidth (BW) and Burst (BT) size as variable parameters because these substantially vary from one service to another. Therefore, these parameters are primarily used in the admission control of BB too. The value of these parameters will eventually determine the total cost of the service used.

4 Implementation & Experiments

4.1 Implementation

Our BB has been enhanced so that it can now negotiate SLOs using our concept of SLO templates. We now explain the data flow and operations performed at various stages in our implementation. In Fig. 3, each box represents a collection of data and an arrow represents operation performed. The box at the tail of the arrow indicates the data before an operation is performed and the box at the head of an arrow represents the data after the operation is over.

First of all, those SLO templates are shown to the consumer for which he/she is eligible (determined by policy). The consumer then performs negotiation for the values of the variable parameters only. This is done using HTML interface designed for this purpose (Fig. 4). The service provider performs conflict detection and consistency check and the result is a **Negotiated SLO (NSLO)**.

At the second step, NSLO is converted into device independent policy to be stored along with other policy rules. At this stage, once again consistency check and conflict detection against other policy rules is performed. In our present implementation, we perform only simple checks, for example, **only one SLO for Premium service may exist for any service consumer at any instance in time**. The consistency and conflict detection checks need to be enhanced as new and complex policy rules are added. For

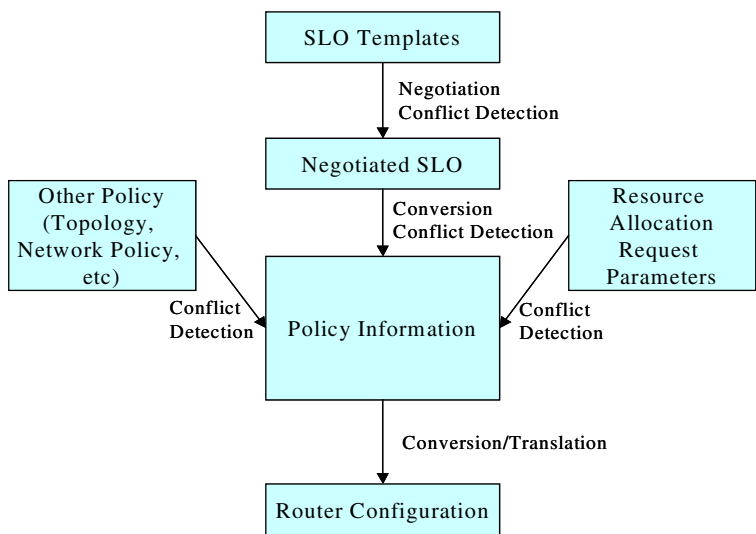


Fig. 4. Data Flow Diagram Showing Data Flow From SLO Templates to Router Configuration

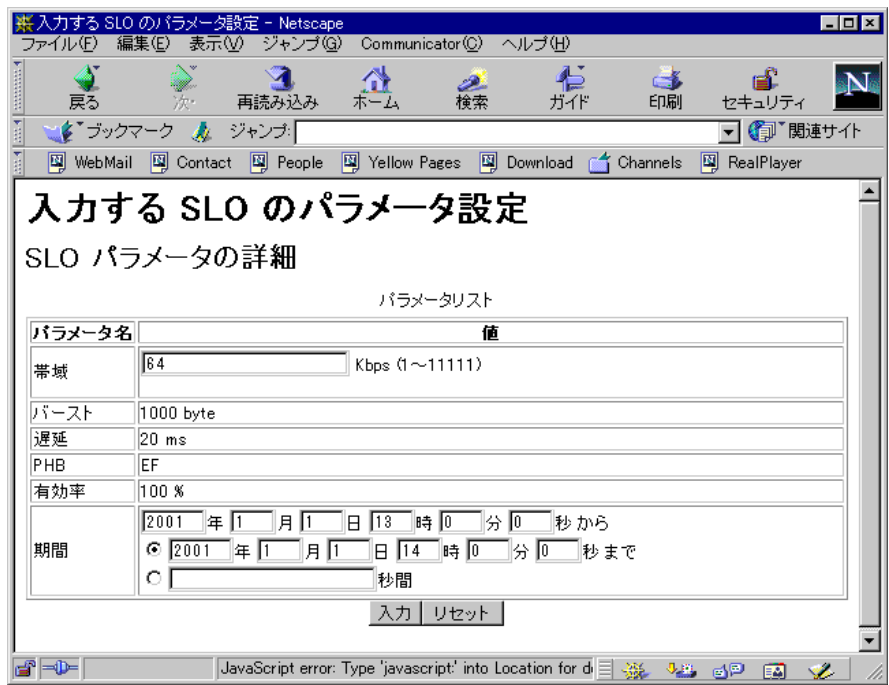


Fig. 4. One page (Japanese) to negotiate variable parameters of SLO templates. Each value box respectively represents bandwidth, burst size, delay, PHB, availability, and time duration

example, one such check would be to *make sure that bandwidth reserved through SLOs for all service-consumers must not be more than a certain % of the available Premium bandwidth*. At stage three, converted policy is again converted into device dependent policy and is then enforced by performing proper router configurations.

4.2 Experiments

To confirm validation of our concept and its implementation, we performed experiments over a LAN, which is divided into three small DiffServ domains (Fig. 5). The main objectives are, to confirm SLO negotiation using SLO templates, to confirm registration of Negotiated SLO (NSLO), and to confirm the translation/conversion of NSLO into final router configurations.

Each DiffServ domain (Domain1, Domain2, and Domain3) in Fig. 5 is managed by a single BB, namely, BB1, BB2 and BB3 respectively. BB1 manages the domain of service consumer and BB2 manages the domain of service provider. The policy provided to BB2 states that BB1 is not financially stable client and thus only *Premium2*, *Premium3* and *Premium4* of all four templates can be negotiated with it. BB1 selects one of these and negotiates values of variable parameters using HTML page shown in Fig. 4. Before BB2 can make a commitment, it performs conversions and conflict detections till an implementable router configuration is derived.

We do not quantitatively measure traffic to check SLO compliance. But because it is a necessary feature and we plan to implement it as a separate module in near future. However, we perform simple qualitative measurement to check SLO compliance.

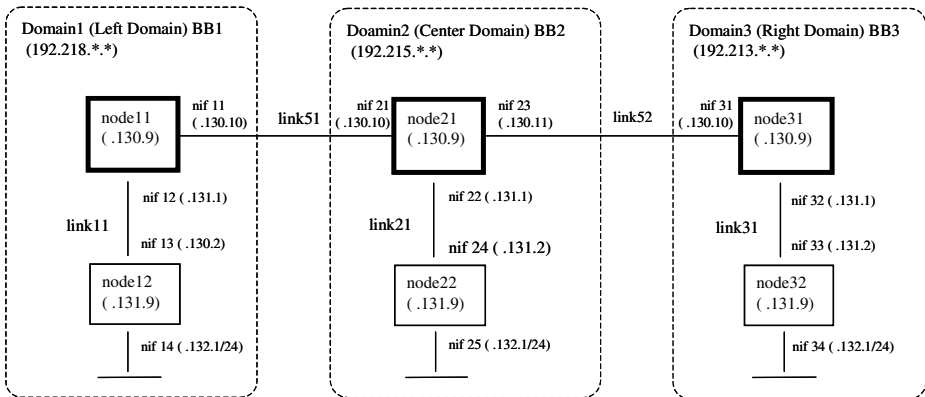


Fig. 5. Three domains network testbed for experiments of SLO registration & negotiation

4.3 Implementation Features and Lessons Learnt

During these experiments many problems were encountered. We briefly write about some important implementation features and lessons learnt during our experiments.

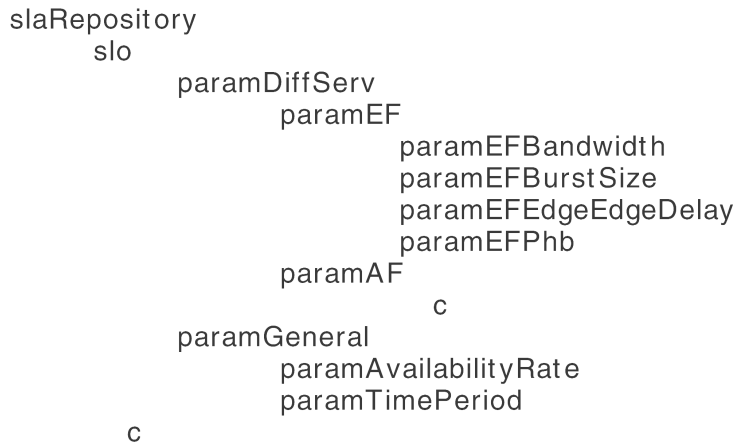


Fig. 6. Portion of Information Model/Schema (Containment Hierarchy) to Store SLOs



Fig. 7. Portion of QoS Policy Information Model (Containment Hierarchy)

4.3.1 SLA Information Model and Schema

To store SLOs, we design an information model and a schema. During our experiments we discovered that SLO mirroring is required which was not anticipated at the beginning. By mirroring, we mean that a copy of NSLO may be possessed by service providers and consumers for verifying NSLO compliance and for accounting purpose. We plan to propose our information model for standardization after making

modifications in the light of feedback from our experiments. Due to the limitation of space and scope, we show only a portion of our information model in Fig. 6.

4.3.2 Policy Framework

We have used a subset of Core Information Model [10] and QoS Information Model [11] proposed in IETF with slight modifications. These modifications are necessary to override the limitations of direct attachment of some object classes. For these experiments we have used only direct attachment and do not use reusable objects. In this concern the scope of some attributes is modified to apply these on the attached object classes as well. A portion of the object classes of QoS Policy Information Model is shown in Fig. 7.

5 Summary and Conclusion

In this paper, we have proposed a new concept of SLO templates, which mainly consists of constant and variable parameters. We propose some SLO templates for the Premium service as an example and to be used to check implementation only. We described the implementation of these templates in our BB and described experiments performed using our implementation.

The concept of SLO templates can be used to easily create many useful services for the mission critical applications of businesses. We expect that with long-term use and popularity, some of the templates will become standardized services, some will be discarded and the others will be employed in local use.

6 Future Work

The followings are the main themes related to the material presented in this paper that we would be focusing on in near future. More research is required before these can be implemented in our BB.

6.1 Enhancement of SLA

The SLO templates proposed in this paper are all related to EF PHB of DiffServ. This service is called as Premium service. In our present proposed SLO templates and their implementation, we do not consider selection of a route. However, it is obvious that QoS routing within a domain and between domains is a necessity from the point of view of optimum resource utilization and from the point of view of policy. We want to investigate the possibility of including route selection parameters in our proposed SLO templates and evaluate their impact using experiments.

6.2 Traffic Measurement for SLA Compliance

Generally speaking, in all business activities, when an SLA is committed between two entities, a proof of the compliance of the SLA needs to be produced by the service provider. The same holds true in case of SLA for network QoS. Traffic needs to be measured and the results need to be provided to the service consumers for the compliance as well as accounting purposes. Note, however, that by traffic measurement we not only mean counting of packets but also detecting faulty links. Traffic measurement is also necessary for the service providers in order to determine the capacity of their future networks.

References

1. K. Nichols, V. Jacobson and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," RFC 2638, July 1999.
2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services," RFC2475, December 1998.
3. V. Jacobson, K. Nichols and K. Poduri, "An Expedited Forwarding PHB," RFC2598, June 1999.
4. J. Wroclawski, "The Use of RSVP with IETF Integrated Services," RFC 2210, September 1997.
5. J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, "Assured Forwarding PHB Group," RFC2597, June 1999.
6. Akira Shirahase, Manzoor Hashmani, Mikio Yoshida, et al., "Design and Deployment of QoS Enabled Network for Contents Businesses," International Conference on Computer Communication 1999, Tokyo Japan, September 14-16, 1999.
7. Koichi Tajima, Manzoor Hashmani and Mikio Yoshida, "A Resource Management Architecture over Differentiated Services Domains for Guarantee of Bandwidth, Delay and Jitter," Information Systems for Enhanced Public Safety & Security (EuroComm2000), Munich Germany, May 17-19 2000.
8. Manzoor Hashmani and Mikio Yoshida "ENICOM's Bandwidth Broker," Saint 2001 Workshops, pp 213-220, Jan 8-12, 2001, San Diego, USA
9. M. Stevens, W. Weiss, H. Mahon, B. Moore, J. Strassner, G. Waters, A. Westerinen and J. Wheeler, "Policy Framework," Internet Draft, Work in Progress, September 1999.
10. B. Moore, E. Ellessen, J. Strassner and A. Westerinen, "Policy Core Information Model -- Version 1 Specification," RFC 3060, February 2001.
11. Y. Snir, Y. Ramberg, J. Strassner and R. Cohen "Policy Framework QoS Information Model," Internet-Draft, Work in progress, November 2000.
12. J. Strassner, E. Ellessen, B. Moore, R. Moats, "Policy Core LDAP Schema," Internet-Draft, Work in progress, March 2001.
13. A. Westerinen, J. Schnizlein, et al. "Terminology," Internet-Draft, Work in progress, March 2001.
14. Danny Goderis, et al. "Service Level Specification Semantics and Parameters," Internet-Draft, Work in progress, November 2000.
15. R. Rajan, E. Celenti and S. Dutta "Service Level Specification for Inter-domain QoS Negotiation," Internet-Draft, Work in progress, November 2000.

Optimal Provisioning and Pricing of Internet Differentiated Services in Hierarchical Markets^{*}

Errin W. Fulp¹ and Douglas S. Reeves²

¹ Department of Computer Science, Wake Forest University, P.O. Box 7311,
Winston-Salem N.C. 27109-7311 USA
`fulp@wfu.edu`

² Department of Computer Science *and* Department of Electrical and Computer
Engineering, N.C. State University, Box 7534, Raleigh N.C. 27695 USA
`reeves@eos.ncsu.edu`

Abstract. Network service providers contract with network owners for connection rights, then offer individual users network access at a price. Within this hierarchy, the service provider must carefully provision and allocate (price) network resources (e.g. bandwidth). However, determining the appropriate amount to provision and allocate is problematic due to the unpredictable nature of users and market interactions. This paper introduces methods for optimally provisioning and pricing differentiated services. These methods maximize profit, while maintaining a low blocking probability for each service class. The analytical results are validated using simulation under variable conditions. Furthermore, experimental results will demonstrate that higher profits can be obtained through shorter connection contracts.

1 Introduction

The Internet continues to evolve from its small and limited academic origins to a large distributed network interconnecting academic and commercial institutions. In this distributed environment, individual users rely on network service providers for network access [5]. Network service providers contract with network owners for connection rights (large amounts over long periods of time), then offer individual users network access (small amounts over short periods of time) at a price. Within this hierarchy, the service provider must carefully provision and allocate (price) network resources (e.g. bandwidth). However, determining the appropriate amount to provision and allocate is problematic due to the unpredictable nature of users and market interactions. Furthermore, provisioning and

^{*} This work was supported by DARPA and AFOSR (grant F30602-99-1-0540). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the DARPA, AFOSR or the U.S. Government.

allocation is more complex with Differentiated Service (DS) enabled networks, since multiple Quality of Service (QoS) classes exist.

It has been demonstrated that resource pricing is an efficient mechanism for resource management, optimal allocations, and revenue generation [2], [3], [4], [6], [7]. However, the majority of these methods are not based on a market hierarchy, and do not consider how to provision resources. Other work has investigated DS resource provisioning [12], but not retail pricing. In contrast, this paper addresses these questions (provisioning and pricing) together within the context of a DS enabled network consisting of hierarchical markets [1]. Goals include **maximizing profit**, as well as maintaining a **low blocking probability**.

The remainder of this paper is structured as follows. Section 2 describes the general design of the hierarchical market economy. Service provider provisioning and allocation strategies are presented in section 3, that maximize profit and reducing the blocking probability. In section 4, the economy is demonstrated under variable conditions, and the monetary advantage of shorter term service level agreements is presented. Finally, section 5 provides a summary of the hierarchical market economy and discusses some areas of future research.

2 The Hierarchical Market Model

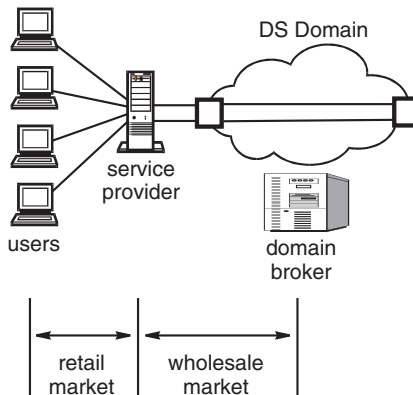


Fig. 1. Example hierarchical market economy consisting of users, service providers, and domain brokers.

As seen in figure 1, the network model is composed of three types of entities (users, domain brokers, and service providers) and two types of markets (wholesale and retail). An individual user, executing an application, requires bandwidth of a certain QoS class along a path. Users may start a session at any time, request different levels of QoS, and have varying session lengths. Furthermore, users desire immediate network access (minimal reservation delay). In contrast, the domain broker owns large amounts of bandwidth (or rights to

bandwidth) and is only interested in selling large DS connections¹ [14]. The service provider plays a very important role in the network economy. Interacting with users and domain brokers, the service provider purchases bandwidth from domain brokers (provisioning), then re-sells smaller portions to individual users (allocation). Buying and selling occurs in two different types of markets: the wholesale market and the retail market.

2.1 Network Resource Markets

In our network economy, service level agreements for future DS connections are bought and sold in the wholesale market. These forward contracts represent large bandwidth amounts over long periods of time [5]. Domain brokers sell contracts for large DS connections, with an associated Service Level Agreement (SLA), across a specific network. An offer specifies the location, delivery date, class q , price g_q , and term. The market then attempts to match a buyer with the seller and a forward contract is created. This is how bandwidth is currently traded in many on-line commodity markets, such as RateXchange and Interxion. If a service provider agrees to purchase a DS connection of capacity s_q , the associated cost is $g_q \cdot s_q$ for the agreed term.

The retail market consists of a service provider selling to individual users, portions of the DS connections purchased in the wholesale market. The price of DS bandwidth will be usage-based, where the user cost depends on the current price and the amount consumed. We will use prices based on slowly varying parameters such as Time of Day (ToD) statistics [9], [10], as seen in figure 2. A day will be divided into T equal length periods of time, where $t = 1, \dots, T$. To provide predictability, these prices (next day) are known a priori by the users via a price-schedule $\{p_{q,t}\}$, where $p_{q,t}$ is the price of class q bandwidth during the t ToD period. The bandwidth of DS connection q is sold on a first come first serve basis; no reservations are allowed. Assume a user requires an amount of bandwidth b_q of service class q for the duration of their session. If the amount is not available at the beginning of the session, the user is considered blocked. However, users who can not afford b_q are **not** considered blocked. Therefore, it is important to price bandwidth to maximize profit as well as maintain a low blocking probability.

3 Optimally Provisioning and Allocating Network Resources

In this section optimal provisioning and retail pricing methods are developed for the service provider, that will maximize profit and reduce the blocking probability. The profit maximization behavior of the service provider is constrained by both markets. To maximize profits, the service provider will seek to make the

¹ Therefore, a session is a small amount of bandwidth (appropriate for a single user or application) compared to a connection.

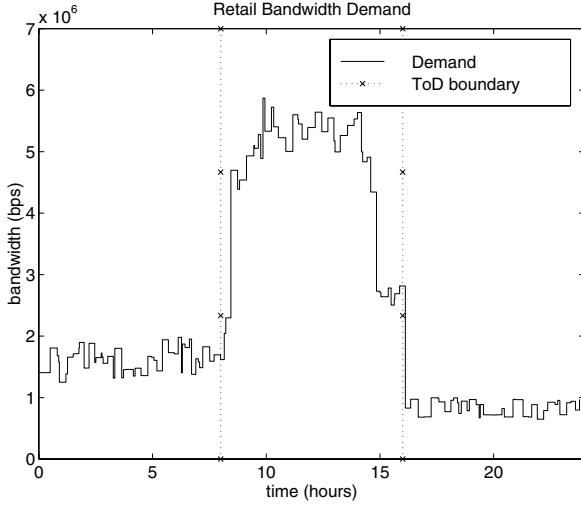


Fig. 2. Example Time of Day (ToD) changes in retail demand.

difference between the total revenue and the total costs as large as possible. The revenue from the retail market for class q during ToD t is $r_{q,t} = p_{q,t} \cdot d_{q,t}(p_{q,t})$. Where $d_{q,t}(p_{q,t})$ is a convex function representing the aggregate retail market demand for service class q during ToD period t at price $p_{q,t}$. As described in section 2.1, the cost of service class q is given from the wholesale market as $c_q = g_q \cdot s_q$. Assume the SLA term for each connection q is N consecutive ToD periods; therefore the supply during ToD $t = 1, \dots, N$ is constant. From the revenue and cost, the profit maximization problem can be written as

$$\max \left\{ \sum_{q=1}^Q \sum_{t=1}^N (r_{q,t} - c_q) \right\} \quad (1)$$

Where profit maximization is over the SLA term. Viewing this as an optimization problem, the first order conditions are

$$\sum_{q=1}^Q \sum_{t=1}^N \frac{\partial r_{q,t}}{\partial s_q} = N \cdot \sum_{q=1}^Q \frac{\partial c_q}{\partial s_q} \quad (2)$$

The left-hand side of equation 2 is also referred to as the marginal revenue, which is the additional revenue obtained if the service provider is able to sell one more unit of DS bandwidth. The right-hand side of equation 2 is referred to as the marginal cost. This is the additional cost incurred by purchasing one more unit of DS bandwidth from the wholesale market. The service provider must purchase (provision) bandwidth from the wholesale market and price bandwidth in the retail market so the marginal revenue equals the marginal cost, as seen in figure 3. If this is done, the profit is maximized and the blocking probability is zero.

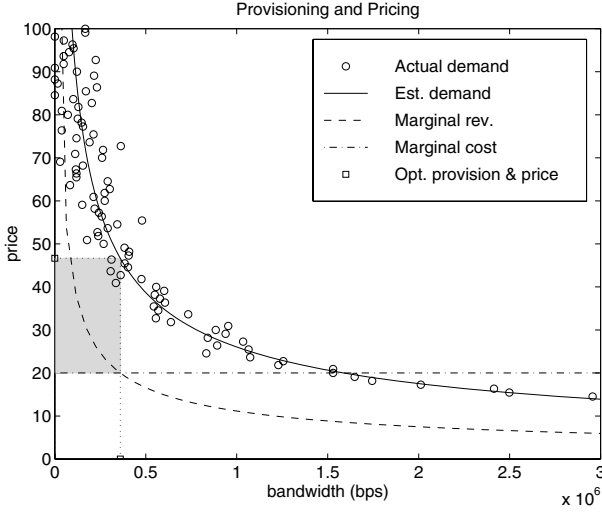


Fig. 3. The service provider seeks the point where the marginal revenue equals the marginal cost. If the optimal provisioning and retail pricing occurs, the amount of profit is given in the shaded area. (Demand data taken from the experimental section)

3.1 Single ToD Wholesale Provisioning and Retail Pricing

In this section, the optimal amount of bandwidth to provision for a single class q for one ToD period t will be determined (the q and t subscripts will be dropped for brevity). Assume the aggregate retail market demand at the retail price p has a Cobb-Douglas form [8],

$$d(p) = \beta \cdot p^{-\alpha} \quad (3)$$

Where β and α are constants describing the aggregate wealth and price-demand elasticity respectively. Price-demand elasticity represents the percent change in demand, in response to a percent change in price². The larger the price-demand elasticity value, the more elastic the demand [8]. The Cobb-Douglas demand curve is commonly used in economics because the elasticity is constant, unlike linear demand curves [11]. This assumes users respond to proportional instead of absolute changes in price, which is more realistic. Therefore, this demand function is popular for empirical work. For example, the Cobb-Douglas demand function has been successfully used for describing Internet demand in the INDEX Project [13]; therefore, we believe this curve is also appropriate for the retail market. Given the aggregate demand function, the revenue earned is,

$$p \cdot d(p) = p \cdot \beta \cdot p^{-\alpha} = \beta \cdot p^{1-\alpha} \quad (4)$$

² Typically elasticity is represented as a negative value, since demand and price move in opposite directions. However, the sign is already incorporated in the demand equation.

Alternatively, the revenue earned by the service provider can be written as,

$$p \cdot d(p) = \left(\frac{\beta}{d(p)} \right)^{\frac{1}{\alpha}} \cdot d(p) = \beta^{\frac{1}{\alpha}} \cdot [d(p)]^{1-\frac{1}{\alpha}} \quad (5)$$

As previously described, the marginal revenue is the first derivative of the revenue equation with respect to the demand; therefore, the marginal revenue is,

$$\beta^{\frac{1}{\alpha}} \cdot \left(1 - \frac{1}{\alpha} \right) \cdot [d(p)]^{-\frac{1}{\alpha}} \quad (6)$$

The cost function for bandwidth is $g \cdot s$ and the marginal cost is g . From equation 2, the service provider maximizes profit when marginal revenue equals the marginal cost.

$$\beta^{\frac{1}{\alpha}} \cdot \left(1 - \frac{1}{\alpha} \right) \cdot [d(p)]^{-\frac{1}{\alpha}} = g \quad (7)$$

Solving for $d(p)$, the optimal amount to provision s_* is

$$s_* = \left[\frac{g}{\beta^{\frac{1}{\alpha}} \cdot \left(1 - \frac{1}{\alpha} \right)} \right]^{-\alpha} = \frac{\beta \cdot \left(1 - \frac{1}{\alpha} \right)^{\alpha}}{g^{\alpha}} \quad (8)$$

During the wholesale market auction, the service provider can use equation 8 to determine the bid amount at the offered price g . Once the auction has closed, the service provider must price bandwidth for the retail market. The optimal retail price p_* is,

$$p_* = \left(\frac{\beta}{s_*} \right)^{\frac{1}{\alpha}} \quad (9)$$

This price causes the demand (equation 3) to equal the supply (equation 8); therefore, the predicted blocking probability is zero (discussed in section 3.3).

The validity of the derived equations can be examined at infinite and unity elasticity. User demand will become very elastic (α approaches ∞), if there is a large selection of service providers (large service provider competition drives profits to zero). In contrast, if the service provider has a monopoly, the elasticity approaches 1 and profits increase [8]. From equations 8 and 9, the optimal revenue under these two extreme cases is as predicted.

$$\lim_{\alpha \rightarrow \infty} \frac{\beta \cdot \left(1 - \frac{1}{\alpha} \right)^{\alpha-1}}{g^{\alpha-1}} = 0 \quad (10)$$

$$\lim_{\alpha \rightarrow +1} \frac{\beta \cdot \left(1 - \frac{1}{\alpha} \right)^{\alpha-1}}{g^{\alpha-1}} = \beta \quad (11)$$

3.2 Multiple ToD Wholesale Provisioning and Retail Pricing

This section considers provisioning for a single class q (the q subscript will be dropped for brevity) over N consecutive ToD periods. These consecutive ToD

periods represent the agreed SLA term from the wholesale market. As described in section 3.1, assume the aggregate retail market demand, during ToD period t at the retail price p , has a Cobb-Douglas form,

$$d_t(p) = \beta_t \cdot p^{-\alpha_t} \quad (12)$$

Where β_t and α_t are constants describing the aggregate wealth and elasticity respectively for ToD period t . The aggregate wealth and elasticity can change from one ToD period to the next. As described in section 3.1 the service provider maximizes profit when marginal revenue equals the marginal cost. Over multiple ToD periods this is

$$\sum_{t=1}^N \left(\beta_t^{\frac{1}{\alpha_t}} \cdot \left(1 - \frac{1}{\alpha_t} \right) \cdot [d_t(p)]^{-\frac{1}{\alpha_t}} \right) = N \cdot g \quad (13)$$

To determine the optimal supply s_* , we must solve equation 13 for $d_t(p)$. However, since the equation is non-linear, a direct solution can not be found. For this reason, gradient methods (e.g. Newton-Raphson) can be used to determine the optimal provisioning amount [15]. Due to the wholesale market auction negotiation time, this calculation can be performed off-line; therefore, convergence time is not critical. Once the auction has closed, the optimal price for ToD period t is,

$$p_{*,t} = \left(\frac{\beta}{s_{*,t}} \right)^{\frac{1}{\alpha_t}} \quad (14)$$

Therefore, in the multiple ToD case, the supply for each ToD period is constant, while the price may vary, as seen in figure 4.

3.3 Retail Market Demand Estimation and Blocking Probabilities

As described in sections 3.1 and 3.2, determining the optimal amount of bandwidth to provision and the retail price requires knowledge of the retail demand curve. However, due to the dynamic nature of the retail market demand can change over time. Such changes may reflect ToD trends, pricing, or the introduction of new technology. For this reason, demand prediction and estimation will be employed [13], where the demand curve parameters (α and β) are estimated using previous ToD measurements. The other goal for the service provider is to maintain a low blocking probability. Based on the optimal provisioning and pricing equations given in the previous two sections, these values will result in supply equaling demand (as seen in figure 3) yielding a zero blocking probability. However, if the estimated demand is less than the actual demand, then the blocking probability will be greater than zero. Therefore, a zero blocking probability depends on accurate demand estimation, which will be demonstrated in the next section.

4 Experimental Results

In this section, the optimal provisioning and pricing techniques described in the section 3 are investigated under variable conditions using simulation. The experiments simulated 6 days, where each ToD was 8 hours in duration (3 ToD per day). The model consisted of 200 users, a domain broker, and a service provider. Users had an elasticity α uniformly distributed between 1.1 and 2.75, and a wealth β uniformly distributed between 1×10^8 and 3.5×10^8 . Furthermore, the demand of each user b_t was uniformly distributed between 0.5 Mbps and 2 Mbps (consistent with multimedia traffic). Each day, users started their sessions at random times using a Poisson distribution with mean equal to the first ToD of that day. This distribution caused the second ToD period of each day to have a high utilization (simulating peak hours). Two separate experiments were performed. The first experiment assumed the SLA term was equal to 6 days, while the second experiment assumed the SLA term was equal to one ToD.

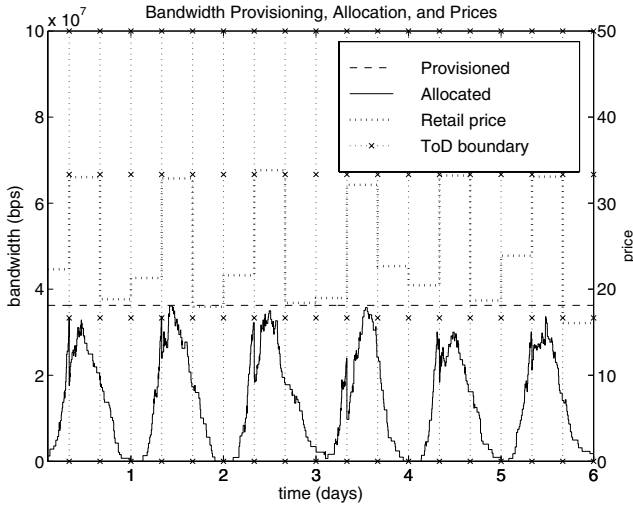


Fig. 4. Retail provisioning, allocation, and pricing simulation results for a six day SLA.

Figure 4 shows the provisioning, allocation, and pricing results, when the SLA term was 6 days. As seen in this figure, the provisioned amount was 36.2 Mbps for the duration of the simulation, while the price per ToD varied from 16.0 to 33.8. Prices during the second ToD of each day were high, since the demand was higher (peak demand). In contrast, the prices for the other ToD periods were low to encourage consumption. The total profit for the simulation was 1.54×10^{13} . The blocking probability was nonzero for ToD periods 5, 8 and

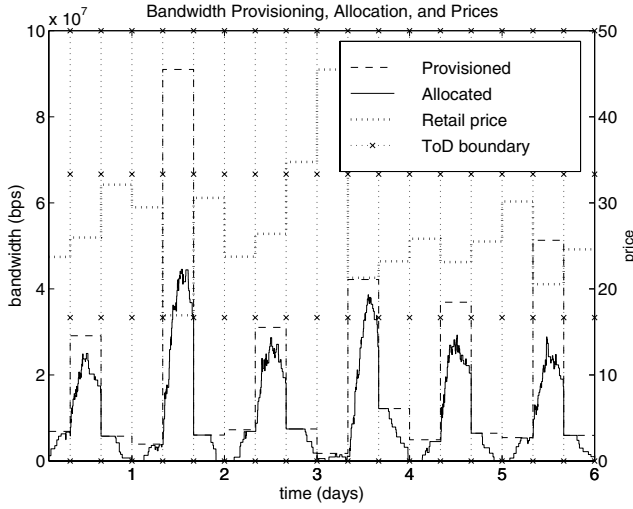


Fig. 5. Retail provisioning, allocation, and pricing simulation results for eighteen SLA's (each term equaled one ToD).

11. During these peak ToD periods, the predicted demand was less than the actual demand.

Figure 5 shows the provisioning, allocation and pricing results when the SLA term was one ToD (18 consecutive SLA's were contracted). In this simulation, the service provider could provision and price bandwidth for each ToD period. The bandwidth provisioned range from 1.7 Mbps to 90.0 Mbps, while the retail price ranged from 16 to 45.5. The total profit was 3.39×10^{13} , over twice as high as the 6 day SLA. Therefore, smaller SLA terms gave the service provider more control (provisioning and pricing), which increased profits. Similar to the other experiment, the blocking probability was nonzero for four ToD periods 4, 7, 10, and 13. Again, this indicates the predicted demand was too small for these periods. However, these were the first ToD periods of the day (non-peak).

5 Conclusions

Network services are typically provided through a hierarchical market economy. This paper introduced a hierarchical economy consisting of two types of markets (retail and wholesale) and three types of entities (service provider, domain broker, and users). Within this market hierarchy, the service provider must carefully provision resources from the wholesale market and allocate resources in the retail market. The service provider seeks to maximize profit and maintain a low blocking probability. However, achieving these objectives is problematic due to the unpredictable nature of the markets. This paper defined optimal buying/selling strategies that maximizes profit while maintaining low blocking probability per

DS connection. These methods rely on retail market estimation to determine the appropriate retail market supply and the retail market price. Simulation results were provided to demonstrate the optimal provisioning and retail pricing methods presented in this paper. The service provider was able to maximize profit given the estimated user demand and the SLA term. Shorter SLA terms were shown to yield higher profits, since the service provider is able to precisely provision based on the ToD statistics. Future work includes investigating sampling procedures and providing retail bandwidth guarantees.

References

1. Y. Bernet, J. Binder, S. Blake, M. Carlson, B. E. Carpenter, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, and W. Weiss. A Framework for Differentiated Services. IETF Internet Draft, February 1999.
2. C. Courcoubetis, V. A. Siris, and G. D. Stamoulis. Integration of Pricing and Flow Control for Available Bit Rate Services in ATM Networks. In *Proceedings of the IEEE GLOBECOM*, pages 644 – 648, 1996.
3. D. F. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini. Economic Models for Allocating Resources in Computer Systems. In S. Clearwater, editor, *Market Based Control of Distributed Systems*. World Scientific Press, 1996.
4. E. W. Fulp, M. Ott, D. Reininger, and D. S. Reeves. Paying for QoS: An Optimal Distributed Algorithm for Pricing Network Resources. In *Proceedings of the IEEE Sixth International Workshop on Quality of Service*, pages 75 – 84, 1998.
5. G. Huston. *ISP Survival Guide: Strategies for Running a Competitive ISP*. John Wiley & Sons, 1999.
6. F. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49:237 – 252, 1998.
7. J. Murphy, L. Murphy, and E. C. Posner. Distributed Pricing for ATM Networks. *ITC-14*, pages 1053 – 1063, 1994.
8. W. Nicholson. *Microeconomic Theory, Basic Principles and Extensions*. The Dryden Press, 1989.
9. A. Odlyzko. The Economics of the Internet: Utility, Utilization, Pricing, and Quality of Service. Technical Report 99-08, DIMACS, Feb. 1999.
10. I. C. Paschalidis and J. N. Tsitsiklis. Congestion-Dependent Pricing of Network Services. *IEEE/ACM Transactions on Networking*, 8(2):171–184, April 2000.
11. D. Reininger, D. Raychaudhuri, and M. Ott. Market Based Bandwidth Allocation Policies for QoS Control in Broadband Networks. In *The First International Conference on Information and Computational Economics*, pages 101 – 110, 1998.
12. N. Semret, R. R.-F. Liao, A. T. Campbell, and A. A. Lazar. Peering and Provisioning of Differentiated Internet Services. In *Proceedings of the IEEE INFOCOM*, 2000.
13. H. R. Varian. Estimating the Demand for Bandwidth. Available through <http://www.INDEX.Berkeley.EDU/public/index.phtml>, 1999.
14. D. Verma. *Supporting Service Level Agreements on IP Networks*. Macmillan Technical Publishing, 1999.
15. S. Yakowitz and F. Szidarovszky. *An Introduction to Numerical Computations*. Macmillan, second edition, 1989.

Adding Interactive Services in a Digital Video Broadcasting Network

Rudolf Jäger

BetaResearch, Betastrasse 1
85774 Unterföhring, Germany
Rudi_Jaeger@betaresearch.de

Abstract. Television (T)-Commerce is not just a buzzword. Consumer and Business demands for Multi-Media services have led to a proliferation of solutions that provide a wide range of services such as digital television, Internet Connectivity, Hidden-Programme Placements and others. Digital Video Broadcasting (DVB) networks which employ the MPEG-2 compression and the transmission standard are used for digital television. Advancements in the network infrastructure as well as in the Set-Top Box technology have led to advanced service models which enables the user to interact actively with the linear presentation model as offered in a broadcasting environment. This paper provides a technological overview of how interactivity can be added to a DVB network. Various scenarios are presented with a different level of interactivity. Accompanied to each scenario the user and the content provider issues are also briefly discussed. The final part attempts to predict a vision on how interactivity in a television environment could look in ten years.

1. Introduction

In the early stages of Digital Video Broadcasting (DVB) its usage was limited to the playout and the transmission of compressed video/audio streams according to the MPEG-2 standard [1]. The user's choice was limited to the selection of video/audio streams. Advancements in the network infrastructure, Set Top Box technology and content authoring techniques have led to advanced service models which enables the user to interact actively with the linear presentation model as offered in a DVB environment. At the same time new business models have been derived offering services such as

- Internet access
- interactive advertisement
- personalised purchasing
- decentralised games

The pure one-to-many broadcasting communication model is augmented by either an one-to-one (single user feedback to a server) or by a many-to-many (internet) communication model. All the services listed above challenge both the user interface design and the technological aspects of the synchronisation between the video stream and some additional data. Since synchronisation of data with video plays a major role

its technical aspects will be discussed in more detail in the following chapters. Chapter two starts with a brief overview of the technical components of a DVB network. The functionality and its technical realisation of the Near Video on Demand (NVoD) Service are explained as an example. In chapter three the various synchronisation means for providing an interactive data service in conjunction with the video stream are discussed. The last chapter provides a vision on possible services and interaction scenarios the user might be faced with in the future.

2. Early Service Models for Digital Video Broadcasting Systems

Digital Video Broadcasting systems are built upon the MPEG-2 standard. The primary function of Set-Top Boxes (STB) supporting early service models was to receive, select and display digital-quality videos. Broadcast video streams are characterised by the fact that video streams are not further accessible after the moment they have been broadcast (limited reception model). Within the video streams specific signalling information in form of tables [2] [3] are multiplexed to provide the STB with signalling information required for synchronisation, tuning and program selection. This information must be available throughout the duration of the program. Due to the fact that anyone can enter the program at any time the signalling information is played out using a data carousel mechanism.

The Near Video on Demand (NVoD) service offering the same program at time shifted beginnings is a typical example of a local interactivity. For this scenario films are played out in an overlapped fashion. The necessary signalling information for selecting the video stream, obtaining the start time and further information about the film is contained in the so-called Event Information Tables (EIT) which are broadcast with a repetition rate of 2 seconds. The EIT contain a so-called reference event and a number of time shifted events equal to the number of time shifted Video streams. The reference event contains information about the name of the movie and a textual description of its content. The time shifted event provides information about the start-time and the duration of the movie. Figure 1 shows a screen shot of an Electronic Program Guide offering interactivity by selecting different time shifted video streams.



Fig. 1. EPG display for selecting different beginnings of a movie

In this model the degree of interaction for the user is limited to the selection of video and audio streams. The selection of a time shifted service is based upon the start time contained in the EIT. The user interacts actively with the user interface of the STB but there is no interaction based upon the content of the video stream possible.

Network Issues

From the network point of view the Digital Video Broadcasting network acts as a large serial disk for the storage of signalling information as well as the video/audio streams. There are no connection and log-on latencies by this kind of communication model. In term of the required bandwidth it can be calculated in advance and shows a relatively static fashion for the duration of the event.

User Interface Issues

In general the user selects one of the feeds to be displayed on the television at a certain instant of time. The design of the user interface should be easy in the sense that a feed selection can be done with very few key strokes e.g. using coloured buttons on the remote control. Also the user interface should present the various starting times to the viewer selecting the one which most recently had started once the user has tuned to that service.

3. Interactivity by Adding Data Service to the Video Service

3.1 Introduction

An enhancement of the pure video selection can be achieved by adding a data stream to a video stream. Regardless of the representation environment or complexity of the data (e.g. Web-Browser/HTML-pages, Image-Viewer/Images) there must be a synchronisation mechanism which bundles the video stream with the additional data. Media convergence as described here automatically calls for media synchronisation. The degree of granularity of the synchronisation can vary from event based to frame based or even based upon an object level. Enriching and synchronising the video stream with data has the full potential for creating attractive consumer services offering new business models. Apart from the unknown social impact of such scenarios legal aspects are another issue which may raise further questions. Those two aspects will not be further investigated in this paper.

Depending upon the service to be offered synchronisation requirements may be either loose which means within seconds, or tight which means frame accurate or even object accurate within a frame.

An example of a loose synchronisation would be an icon that appears few seconds after the start of an event allowing further interaction. A loose synchronisation is typically event driven relaying upon the DVB signalling information mechanism and does not require time stamps carried within the additional data stream. Within this signalling information an absolute time could be carried for the synchronisation. The next degree of granularity concerns the frame based data synchronisation. This method requires the presence of time stamps which are correlated to the video stream time line. An example of a tight synchronisation could be links that are connected to

frames (scene) or even follow objects displayed on the television screen. Since the synchronisation aspects play a major role in providing interactivity in a DVB environment, three different synchronisation mechanism will be discussed in more detail in the following subchapters.

3.2 Event Based Synchronisation of Data Streams

For this kind of scenario the additional data are synchronised upon the occurrence of a certain event of a program. At BetaResearch a signalling protocol which enables the synchronisation of data with the video stream of a granularity of one second has been developed [4]. Specific tables are broadcast which contain trigger events (start time and duration) with an absolute time base. Thus synchronisation within an event is possible. In other words the broadcast video and data are coupled via the absolute time base. Figure 2 shows an example of an user interface where the pure video is augmented with web based data (e.g. HTML pages) that is related to the video content using the BetaResearch signalling protocol. Normally the video occupies the whole screen. The availability of additional data is indicated by an icon. If the user clicks on the icon the video is displayed in a smaller area of the screen, a browser window opens displaying related information to the video. The user can obtain further information of the car/driver by pressing a coloured button on the remote control. The additional data (e.g. HTML pages) is either broadcast (walled garden) or could be retrieved from a server administered by the service provider. For the walled garden approach the additional data is broadcast by means of a data carousel according to the schedule of the video stream or in advance for possible pre-processing at the STB. The content of the data carousel may vary during the event.



Fig. 2. Display example of a Video-Web Synchronisation

Network Issues

A scheduling system at the head end site is necessary to generate and playout the tables (signalling information) as well the actual data (HTML pages) according to the program schedule. Authoring of the video content in advance is not necessary. The multiplexer pre-allocates either sufficient bandwidth for the data carousel (static bandwidth allocation) or a bandwidth manager controls and allocates the bandwidth among the sources (video, data carousel). The bandwidth manager becomes necessary if the data carousel varies its size over the duration of the event ensuring an optimal utilisation of the transponder bandwidth. Figure 3 shows an example of an architecture of a DVB satellite network offering interactivity by means of data carousel and on-demand data access via the return channel. If a satellite network is used for the delivery of both video streams and on demand requested data (e.g. HTML pages) a bandwidth allocation scheme is needed to avoid possible bandwidth interference among the carousel data, on demand data streams and the video/audio broadcast streams. In [5] a bandwidth allocation scheme has been proposed which controls the bandwidth requirement of the response data (over-air delivery) by adjusting the TCP window size.

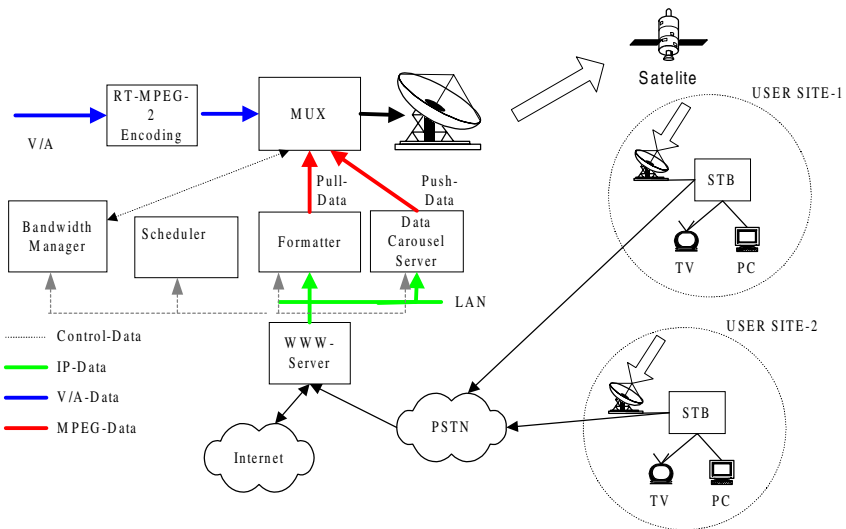


Fig. 3. DVB-S network architecture supporting interactive data services

Content Provider Issues

The walled garden approach provides a more secure environment for the service and content provider thus easier to control than the free internet approach. This will also lead to more secure revenue streams. The content provider must be aware of the presence of a linear model (video stream) and to some degree of the interactive component. In order to attract the user with this kind of interactivity, a strong correlation of the video and the data taking into account the user's profile should be ensured. For example a video showing a holiday resort might be augmented with

additional information covering many aspects such as evening events, tours offered or the temperature of the pool. Depending upon the profile of the user (hobbies, cuisine etc.) the additional data is filtered and presented in a way that it matches the user's expectations. Thus it is possible to provide the user with the most valuable information that ensures highest user satisfaction.

User Interface Issues

The user is asked for a different way of behaviour and attention depending upon the screen (video or data interaction) selected. Navigation has to support the dual interaction. This dual screen approach (video and interactive data) challenges the user interface design by supporting different interaction spaces. The TV viewer enjoys the leisure whereas the data interactive screen asks for an active involvement by the user. Also the one-to-many communication model (video and carousel data) is enriched by an one-to-one communication model (user – server request).

Within this chapter synchronisation mechanism based upon an event level were discussed. In the following chapter content based synchronisation means will be further discussed.

3.3 Frame Based Synchronisation of Data

Due to the temporal nature of video and the fact that there are 25 frames per second (50Hz systems) the synchronisation on a frame based level becomes a difficult task. In the worst case a synchronisation and an interactivity point may be defined for each frame. A shot which consists of series of frames or a scene consisting of hundreds of frames requires more relaxed synchronisation constraints. In order to enable frame based synchronisation with additional data two basic approaches are possible. Both approaches require an authoring process of the content.

3.3.1 Utilising MPEG-2 Header Parameters

The first approach utilises the possibility to add data in the MPEG-2 header ("user data field" parameter). The MPEG-2 header of a compressed video stream is modified by adding a very limited amount of meta data during the authoring process.¹ The meta data refers to the actual data to be synchronised with the video stream. At the STB the decoder could be programmed in such a way that by discovering the meta data further actions are being taken.

3.3.2 Using the MPEG-2 Timeline

The second approach outlined in [6] focuses on the MPEG-2 timing model which means that the data stream maintains the same time line as the MPEG-2 program stream. Technically speaking the data is synchronised to the MPEG-2 source. It refers to the same program clock reference timeline as the MPEG-2 video stream and thus is contained in the same Program Map Table as the video program stream. Depending upon the size of the data to be synchronised with the video two technical implementations are possible. Either the actual data with some meta data attached is

¹ Advanced Cameras/Encoder equipment may allow to add meta data on line.

synchronised to the video as proposed in [6]. Or an extension offering more flexibility in terms of linking the content (frame) to different data would be to use meta data as a reference to the actual data synchronised with the video stream. This approach also offers more flexibility if the data is transferred over a network which is different than the video stream. If the meta data approach is used it must be ensured that the actual data are received and decoded in time at the STB. The necessary timing relationship between the video (frame) and the data is established during the authoring process taking place ahead of time. The outcome is a parameter duple comprising the Presentation Time Stamp (i.e. a pointer to the video frame) and meta data.

Figure 4 depicts an example of synchronising two data streams with a video stream at t_1 and t_2 by means of the meta data synchronisation approach using the MPEG-2 time line. The meta data synchronised with the video stream via the presentation timestamps contain a link to the location of the actual data depicted in figure 4 as `addinfo_1` and `addinfo_2` which could be e.g. HTML-pages. In addition further information relevant to the evaluation of the user's profile may be contained. The data e.g. HTML pages processed at the STB have to be transmitted and cached ahead of time in order to guarantee frame accuracy.

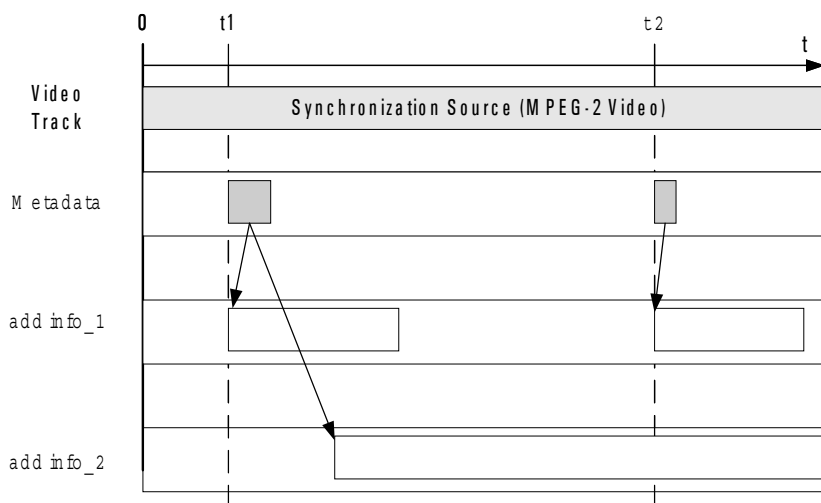


Fig. 4. Synchronisation of video and data by means of meta data

3.4 Object Based Synchronisation of Data

The solution with the most granularity concerns the synchronisation of data with objects within frames or series of frames (shots). At this point it is worthwhile to spend a few words on the MPEG-2 compression algorithm. Transform based compression algorithm like the MPEG-2 standard which decomposes images into square pieces (picture > Macroblocks > blocks) does not consider the image composition by means of objects. Due to this kind of compression the object based

synchronisation becomes difficult. Improved algorithm which model and compress the object motion rather than trying to encode arrays of blocks of pixels would be more appropriate. The MPEG-4 standard is an attempt to enable the possibility of object based synchronisation on an object level of the image. In order to synchronise data upon an object level in a MPEG-2 compression environment, enhanced scene analysis methods which are part of the authoring process need to be employed. Generally an object may move along a certain area of the screen for one shot or scene. If someone wants to synchronise the data with an object, methods of object tracking mechanism are needed. In [7] an object segmentation algorithm was described which is applied to uncompressed video allows a classification of every pixel in every frame of a video sequence. The results are data base entries associating marked objects with some action to be performed when selected by the viewer.

Network Issues

If the transfer of the data (meta data or actual data enabling interaction) is done on the same network as the video is distributed, then either multiplexing is needed or the data is transferred within pre-allocated stuffing packets. If the former one is used the Program Clock Reference needs to be recalculated. If the latter one is used, a control application must be developed ensuring that the data are added synchronously to the video stream occupying the stuffing packets. Also the occurrence of the additional data must be signalled in the relevant MPEG-2 tables allowing the DEMUX to trigger on this data. The transfer of the actual data by a different delivery network would be an additional possibility.

Content Provider Issues

The linear model of video presentation is still left intact by allowing interactivity. The strength of allowing interactivity based upon frame or object of the video offers new business opportunities. It enables to promote new business models such as *in-program sponsorship* (e.g. links providing further information might be sold), *hidden product placement* (e.g. links hidden behind objects which offer buying options might be sold) or personalised advertisement targeting all with the potential to create additional revenue streams for the content as well as service provider. An example of *in-program sponsorship* could be to sell links to further information triggered upon frames or objects. An example of *hidden product placement* could be to offer buying options hidden behind frames, scenes or objects. In other words it enables broadcaster to offer additional program sponsorship deals. If you offer a new business you automatically ask for methods to quantify the new service in terms of what kind of charging model is acceptable for all components appearing in the value chain. Apart from the business aspects legal issues may also arise with this kind of new service models, e.g. is a hidden information to an add legally treated like a real add or not simply because the user is asked to turn the hidden into a real one?

User Interface Issues

TV program-metaphor based interfaces usually require less actions than well-known Web based user interfaces. Once a TV user has tuned to a program he/she only has to watch and listen. Offering clickable icons which are content driven or clickable

objects the user interface needs to be adapted to content based interactive TV. Several user interface design questions have to be tackled e.g.

- "How is it indicated on the screen that there is a hidden information behind an object not annoying the viewer and interfering the plot too much ?"
- "What is the response for the user on the screen once the user has selected an icon ?"

The degree of disturbance the user is willing to accept for the added value is another question. A user study is necessary to evaluate both feature usage and overall experience in order to answer those and many more questions.

Since prototypes of STB and head end systems supporting event and frame synchronised data services are already available interactive advertisement or hidden program placement services will hit the market within the next two years. In the following chapter an attempt has been made to predict the evolution of interactive services in a digital video environment.

4. Future Interactive Service Scenarios

4.1 Intelligent Agents

While the business moves towards globalisation the interactivity space will gravitate towards unification considering the personal profile of the user/viewer. The huge amount of redundancy of content conveyed to the user (content consumer) will be filtered by intelligent agents and present the relevant interactivity space. The agent will learn about the users' preferences which are stored in a data base and updated in an adaptive way. Together with the accompanied meta data the degree and the space of interactivity are tailored to the user. Voice and fingerprint recognition or even other means will be used for authentication.

This service model still relies upon the linear limited reception model for the video stream but the interactive data related service offers a non linear component. The one-to-many communication model will be augmented by an one-to-one or even by a many-to-many communication model, e.g. every user's hard disk may be accessed by every other users. This certainly requires more powerful STB, but as the processor power doubles every two years such STB may appear on the market within the next three to five years. In addition to this video technologies will be developed which will partially replace the linear reception model with a non linear presentation model. For example the viewer gets the option to compose and play a movie according to his/her favourites still having a plot. The services adapt to the action of the viewers. A certain degree of responsiveness of the viewer is taken into account for the composition of the program or interaction space for the viewer.

4.2 Meaning of Audio Visual Information

The increasingly pervasive role that audio visual sources are going to play in the future of our life will make it necessary to develop new forms of representing visual

audio information considering also e.g. the meaning of the information conveyed to the user. At the user's side the mood of the viewer or the environment also needs to be taken into account offering a certain space of interactivity for a specific type of content ("Emotional Devices"). The current and upcoming standards like the object based MPEG-4 standard are not suitable and need to be improved. In order to realise such services the content must be augmented with physical and psychological information which have a reflection on the way human beings' behave, e.g. depending upon the mood, emotion and the environment of the viewer, movies containing scenes and a plot that fits best to the viewer will be for selection at the program guide. Summarising it, new audio visual presentation means affecting both the content production phase as well as the user's interaction behaviour have to be researched in the long term range. This kind of service may reach the customer in about six to ten years.

Acknowledgement

Special thanks to Ingo Barth for its technical review and contribution to the BetaResearch signalling protocol issue and to Frank Lonczewski for its valuable discussions on the user interface issues. Thanks also go to David Gillies, Melanie Kogler, Andreas Penka and to Andreas Waltenspiel for their overall review improving the readability of the paper. Thank you to Matthias Kautzner and Matthias Zink for the graphical preparation of the pictures.

References

1. ISO 13818-1 Information Technology – Generic Coding of Moving Pictures and Associated Audio: Systems, November 13, 1994
2. ETR 300 468 Specification for Service Information in DVB Systems, January, 1977
3. ETR 211 Guidelines on Implementation and Usage of Service Information (SI), August, 1997
4. BetaResearch Carousel Signalling Protocol, BetaResearch Internal Specification; Version 1.4; 29.05.2000
5. Jaeger, R., Managing Video/Audio- and Asynchronous Data Streams in a DVB Network; in Proc. IEEE Globecom 08 – 12 November 1998; Sydney
6. Brunheroto, J. et al; Issues in Data Embedding and Synchronization for Digital Television; in Proc. IEEE International Conference on Multimedia and Expo 30.July – 2 August 2000; New York; (ISBN 0-7803-6536-4)
7. Bove M. et al; Hyperlinked television research at the MIT Media Laboratory; IBM System Journal, Vol. 39; NOS 3&4, 2000

A Discrete-Time Queuing Analysis of the Wireless ATM Multiplexing System

M. Mehmet Ali, X. Zhang, and J.F. Hayes

Department of Electrical and Computer Engineering
Concordia University
1455 De Maisonneuve Blvd. West, Montreal, Quebec, Canada, H3G 1M8
mustafa@ece.concordia.ca

Abstract. In this paper we analyze a wireless channel model which is subject to service interruption either because the channel is not available or the server is serving to the other users. The wireless channel is modeled by a Markov chain with two states corresponding to high and low error states, respectively. It is assumed that the channel is slotted in time and no transmission is possible during the high-error state. The traffic generated by a mobile user is modeled as the superposition of independent two states *On-Off* Markov sources, which are statistically independent of the server. A source generates packets only during an *On* state each of which fits to a channel slot. A discrete-time queuing analysis derives the probability generating function (PGF) of the queue length under the assumption of an infinite buffer. From the PGF, mean queue length, mean delay and approximate finite buffer overflow probabilities are calculated.

1 Introduction

Our study is motivated by packet transmission over wireless channels. The stated goal of wireless networks is ubiquitous communication, i.e., digital connectivity of any type, at anytime and anywhere. First and second generation wireless networks currently provide support for circuit-switched voice services as well as low-rate circuit-switched and packet-switched data services. This has led to demands for broadband services and the design of third-generation wireless systems, e.g., IMT2000. Wireless ATM is viewed as a general solution for the third-generation wireless networks capable of supporting multimedia [1] and it is a direct result of the success of ATM on wired network. ATM was originally designed for bandwidth rich and nearly error-free medium (optical fiber), and it effectively trades off bandwidth for simplicity in switching. In contrast, in the wireless environment, the radio channel is bandwidth-limited and the bit-error-rate (BER) is time varying, an effect of distinct propagation phenomena such as multi-path fading, shadowing, path loss, noise and interference from other users. The time-varying channel poses challenges to the design of wireless ATM. It is therefore important to obtain good understanding of the performance (e.g. loss and delay) of employing ATM over burst-error wireless channels.

Early work in modeling wireless channel focused on the stochastic behavior of the channel at the physical layer, measured by received signal strength or bit-error-rate.

Such physical layer models cannot be directly used to evaluate higher-layer network performance, such as queuing delay and loss probability. For example, a few bit errors within a packet, which is the basic data unit at the wireless link layer, will cause the entire loss of the packet. It is therefore imperative to develop packet-level wireless channel models, which can be used by network engineers to simulate and analyze the higher-layer performance of wireless networks. The most commonly used model is Gilbert-Elliott model, which uses a two-state Markov chain to describe the channel [2]. In this channel model, each state corresponds to a specific channel quality that is either noiseless or totally noisy, as "Good" state or "Bad" state respectively. Previous studies show that these Markov chains provide a good approximation in modeling the error behavior at the packet level in wireless fading channels.

The system under consideration is a wireless ATM network cell that consists of a base station and a number of mobile users. The base station provides the interface to the wired ATM network, thus the wireless ATM network extends the reach of the wired ATM network to the mobile users. Each mobile user may generate multi-media traffic which will be buffered at the mobile user. It will be assumed that the arrival process to each user may be modeled as a superposition of mutually independent *On/Off* sources since this type of sources have been widely used to model broadband traffic including voice and video [3]. The two states of the binary sources are "On" and "Off", respectively, with geometrically distributed sojourn times, measured in slot times. In the "On" state, the source generates at least one packet. This work models the transmission of one of the mobile users in this wireless ATM system. Since the wireless channel will be shared by a number of users, periods of channel unavailability may be due to the transmission of signals by other users as well as periods of fade.

There is a significant amount of work on multiplexing in ATM. In [4], the fluid approximation method was applied to analyze an infinite buffer ATM multiplexer, which is loaded with the superposition of statistically independent and identical *On/Off* sources. A Matrix-Analytic approach is used for a discrete-time queuing analysis of an ATM multiplexer in [5]. In [6], a model with binary *On/Off* Markov sources was presented and a discrete-time queuing analysis using a generating function approach was developed. The functional equation describing the ATM multiplexer derived in [6] was solved in [7] and the PGF of the queue length was obtained. The servers in the queuing analyses mentioned above are all assumed to be deterministic, which means the server is always available, corresponding to the situation in wired ATM network.

The salient feature of our model is service interruptions. This is a topic that has a good deal of prior work both on continuous and discrete-time queuing systems with server interruptions [8, 9]. A discrete-time analysis of a system with a general service interruption process and uncorrelated arrival process has been presented in [9]. Since, in this paper we assume a two-state Markov model for the server, this is a special case of the server model in [9] corresponding to geometric On-times and geometric Off-times. On the other hand, we assume more complicated correlated arrival process as opposed to the uncorrelated arrival process in [9]. [10] studies a multiplexer operating in a two-state Markovian environment in which the arrival process and the availability of the output channel in each state is different. In that work, the arrival process and the server interruption process are not independent and they depend on the same two-state Markovian environment. In contrast, we have a more general arrival process resulting from m homogeneous, independent two-state Markov sources. Further, our

arrival process is independent of the server interruption process, which is more appropriate to the applications environment under consideration. Thus the results of this work advance the state-of-the-art on discrete-time queuing systems when both arrival and server interruption processes are correlated.

2 Analytical Model

In this paper, we model a mobile user in a wireless ATM multiplexing system as a discrete-time queuing system with infinite queue length and a single stochastic server. The time-axis is divided into equal slots and packet transmission is synchronized to occur at the slot boundaries. It is assumed that a packet can not leave the buffer at the end of the slot during which it has arrived and that a packet transmission time is equal to one slot.

The arrival process of the system consists of m mutually independent and identical binary Markov traffic sources, each alternating between *On* and *Off* states. We assume that during a slot an *On* source generates at least one packet, while during an *Off* slot the source generates no packet. State transitions of the sources are synchronized to occur at the slots' boundaries. A transition from *On* to *Off* state occurs with probability $1-\alpha$ at the end of a slot, thus the number of slots that the source spends in *On* state is geometrically distributed with parameter α . Similarly a transition from *Off* to *On* state occurs with probability $1-\beta$ at the end of a slot, and the number of slots that the source spends in *Off* state is geometrically distributed with parameter β . When α and β are high, packets have tendency to arrive in clusters, alternatively when α and β are low, then packet arrivals are more dispersed in time.

The server is also modeled as a two-state Markov chain, which alternates between *Good* and *Bad* states. We assume that during a slot, if the server is in *Good* state, it will transmit a packet if there are packets in the queue, while in *Bad* state the server will not transmit any packets whether there are packets in the queue or not. A transition from *Good* to *Bad* state occurs with probability $1-\gamma$ and a transition from *Bad* to *Good* state occurs with probability $1-\sigma$. As a result, the lengths of the *Good* and *Bad* periods are also geometrically distributed with parameters γ and σ , respectively.

Now let us make the following definitions,

m = number of sources in the system.

i_k = length of queue at the end of slot k .

n_k = state of the server during slot k ('1' for *Good* and '0' for *Bad*).

a_k = number of *On*-sources during slot k .

b_k = number of packets that arrive during slot k .

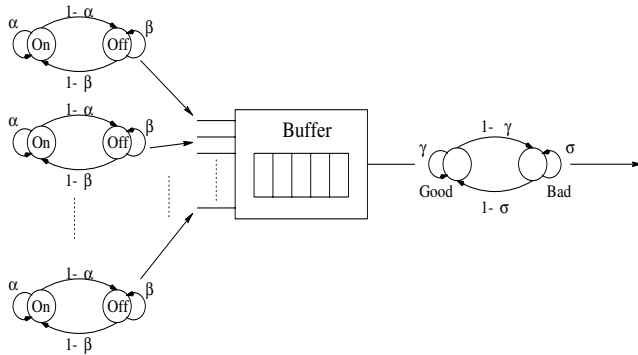


Fig. 1. Model of a wireless ATM multiplexing system

$f_{j,k}$ = number of packets generated by the j 'th *On*-source during slot k .

We assume that the $f_{j,k}$'s are independent identically distributed from slot to slot during an *On* period with PGF $f(z)$ and average $\bar{f}$.

The queuing system under consideration can be modeled as a discrete-time three-dimensional Markov chain. The state of the system is defined by the triplet (i_k, n_k, a_k) .

Let $Q_k(z, r, y)$ denotes the joint probability generating function of i_k , n_k and a_k , i.e.,

$$Q_k(z, r, y) = E[z^{i_k} r^{n_k} y^{a_k}] = \sum_{i=0}^{\infty} \sum_{l=0}^1 \sum_{j=0}^m z^i r^l y^j p_k(i, l, j). \quad (1)$$

where $p_k(i, l, j) = \Pr(i_k = i, n_k = l, a_k = j)$. We have determined the steady-state joint PGF, $Q(z, r, y)$, using a transform analysis details of which may be found in [11]. Next, we present the steady-state PGF of the queue length, $P(z)$, from $P(z) = Q(r, z, y) \Big|_{r=1, y=1}$,

$$P(z) = [(1 - \sigma)p(0, 0, 0) + \gamma p(0, 1, 0)](z - 1) \sum_{i=0}^m \binom{m}{i} \tilde{C}_1 \lambda_1^i (\tilde{C}_2 \lambda_2)^{m-i} \left[\frac{\tilde{C}_3}{z - \lambda_1^i \lambda_2^{m-i} \lambda_3} + \frac{\tilde{C}_4}{z - \lambda_1^i \lambda_2^{m-i} \lambda_4} \right] \quad (2)$$

where,

$$\lambda_{1,2} = \frac{\beta + \alpha f(z) \mp \sqrt{(\beta + \alpha f(z))^2 + 4(1 - \alpha - \beta)f(z)}}{2}. \quad (3)$$

$$\lambda_{3,4} = \frac{\gamma + \sigma z \mp \sqrt{(\gamma + \sigma z)^2 + 4(1 - \gamma - \sigma)z}}{2}. \quad (4)$$

$$\tilde{C}_{1,2} = \frac{1}{2} \mp \frac{2(1-\beta-\alpha)f(z) + \beta + \alpha f(z)}{2\sqrt{(\beta + \alpha f(z))^2 + 4(1-\alpha-\beta)f(z)}}. \quad (5)$$

$$\tilde{C}_{3,4} = \frac{1}{2} \mp \frac{2(1-\gamma-\sigma)z + \gamma + \sigma z}{2\sqrt{(\gamma + \sigma z)^2 + 4(1-\sigma-\gamma)z}}. \quad (6)$$

$$p(0,1,0) = \frac{1-\sigma}{2-\gamma-\sigma} - \frac{m(1-\beta)\bar{f}}{(2-\alpha-\beta)}. \quad (7)$$

$$p(0,0,0) = \frac{1-\gamma}{2-\gamma-\sigma} - \frac{m(1-\beta)(1-\gamma)\bar{f}}{(1-\sigma)(2-\alpha-\beta)}. \quad (8)$$

The probability that the system is busy is given by,

$$\rho = \frac{m(1-\beta)\bar{f}}{2-\alpha-\beta} \cdot \frac{2-\gamma-\sigma}{1-\sigma}. \quad (9)$$

and the stability condition of the system requires that $\rho < 1$.

Next, we present the mean queue length, $\bar{N}$, which may be obtained by differentiating the equation (2) with respect to z and then substituting $z = 1$,

$$\bar{N} = P'(z) \big|_{z=1} = \frac{S''(1)}{2(1-S'(1))} + M'(1) \quad (10)$$

where,

$$S'(1) = \frac{m\bar{f}(1-\beta)}{2-\alpha-\beta} + \frac{1-\gamma}{2-\gamma-\sigma} \quad (11)$$

$$M'(1) = m \left[\frac{\bar{f}(1-\beta)}{2-\alpha-\beta} + \frac{\bar{f}(1-\beta)(1-\alpha-\beta)}{(2-\alpha-\beta)^2} \right] + \frac{(1-\gamma)(1-\gamma-\sigma)}{(2-\gamma-\sigma)^2} \quad (12)$$

$$\begin{aligned} S''(1) = & \frac{m(m-1)(1-\beta)^2(\bar{f})^2}{(2-\alpha-\beta)^2} + \frac{2m(1-\alpha)(1-\beta)(\alpha+\beta-1)(\bar{f})^2}{(2-\alpha-\beta)^3} + \frac{m(1-\beta)f''(1)}{2-\alpha-\beta} \\ & + \frac{m(1-\beta)(1-\gamma)\bar{f}}{(2-\alpha-\beta)(2-\gamma-\sigma)} + \frac{2(1-\gamma)(1-\sigma)(\gamma+\sigma-1)}{(2-\gamma-\sigma)^3} \end{aligned} \quad (13)$$

Finally, from the Little's result, the mean packet delay in number of slots is given by,

$$\bar{d} = \frac{\bar{N}(2-\alpha-\beta)}{m(1-\beta)\bar{f}} \quad (14)$$

As may be seen, we present completely determined closed form expressions for the PGF and the mean queue length. We note the mean queue length may be very easily calculated from (10), since the expression is in terms of the system parameters

3 Numerical Results

In this section we present some numerical examples regarding the results of this paper. The objective is to study the effect of different wireless link error characteristics on the behavior of the ATM multiplexing system. The different wireless channel characteristics are represented by the parameters γ and σ , which $1-\gamma$ and $1-\sigma$ are the probability of transition from *Good* to *Bad* state and the probability of transition from *Bad* to *Good* state in each time slot, respectively. As stated earlier on, in the bad state channel is not available either due to fading or server serving other users. It is assumed that each *On* source generates a single packet during a slot, i.e. $f(z) = z$.

In Figures 2 - 4, we plot the steady-state mean of the queue length as a function of the number of sources, m , with mean good and bad durations as parameters, which are given by $1/(1-\gamma)$ and $1/(1-\sigma)$ respectively. As may be seen, for the same number of sources, different mean queue lengths are obtained for different wireless link error rates. From Figure 2, as the mean bad duration increases from 0 to 40 slots, the corresponding steady-state mean of the queue length also increases. This is expected, since the longer the bad duration, the longer that the channel is in *Bad* state, during which the queue builds up. In Figure 3, the ratio of the means of bad and good durations has been kept constant. As may be seen for lower values of the number of sources in the system the mean queue length does not vary much with the individual values of the mean good and bad durations when their ratio is constant. In also Figure 4, we present mean queue length at constant ratio of the means of bad and good durations but with transmission rate as a parameter. As expected, with increasing transmission rate the mean queue length drops.

Figures 5 presents the mean packet delay as a function of the traffic load ρ with the number of sources in the system, m , as parameters. The results are given for the mean bad duration of 2 slots, mean good duration of 20 slots, and the number of sources, $m = 2, 5, 10$ and 100. As may be seen, for any given load, an increase in the number of sources leads to a rise in the mean packet delay.

From Figures 2 - 5, we also note that under heavy loading, there is a sharp increase in the mean queue length and the mean packet delay. This is the reason why we did not keep that part of the curves corresponding to loads higher than 0.8.

Figure 6 presents the approximate probabilities of buffer overflow under different buffer sizes. The approximate probabilities of overflow corresponds to the probabilities that the queue length will be greater than the chosen buffer size in the infinite queue length system that we have studied. These probabilities may be obtained by performing a Taylor series expansion of the PGF of the queue length and summing up the appropriate coefficients which may be simply done by using any of the available symbolic softwares. As expected, for any given level of traffic, the probability of overflow decreases as the mean bad duration decreases. Unfortunately,

because of the finite precision problems we could not obtain probabilities of buffer overflow for systems with larger buffer sizes.

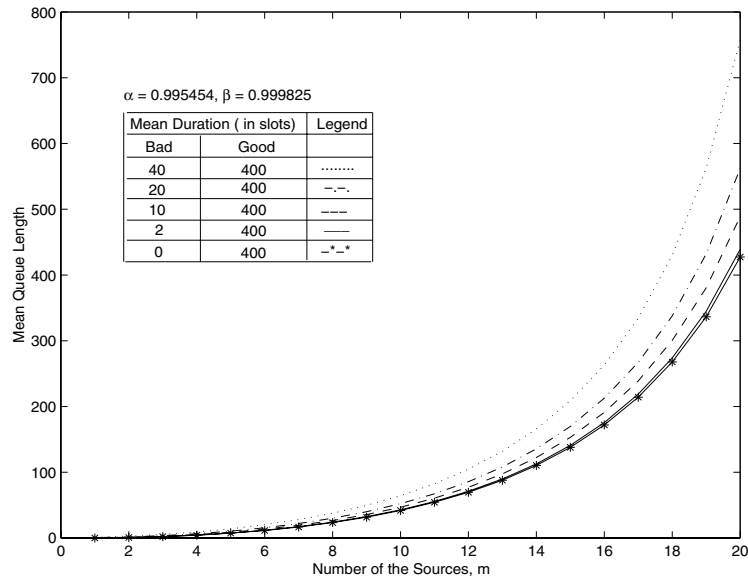


Fig. 2. Steady-state mean of the queue length versus the number of sources, m , when the mean good duration is 400 slots

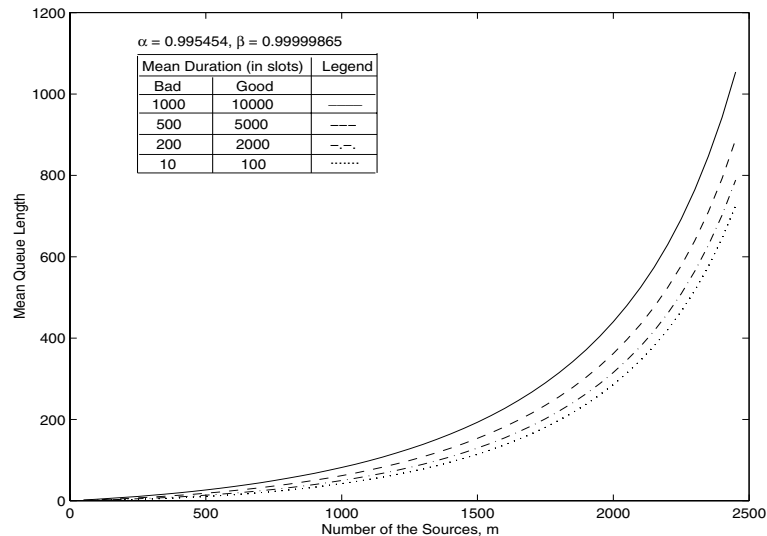


Fig. 3. Steady-state mean of the queue length versus the number of sources, m , for constant ratio of mean good to mean bad duration

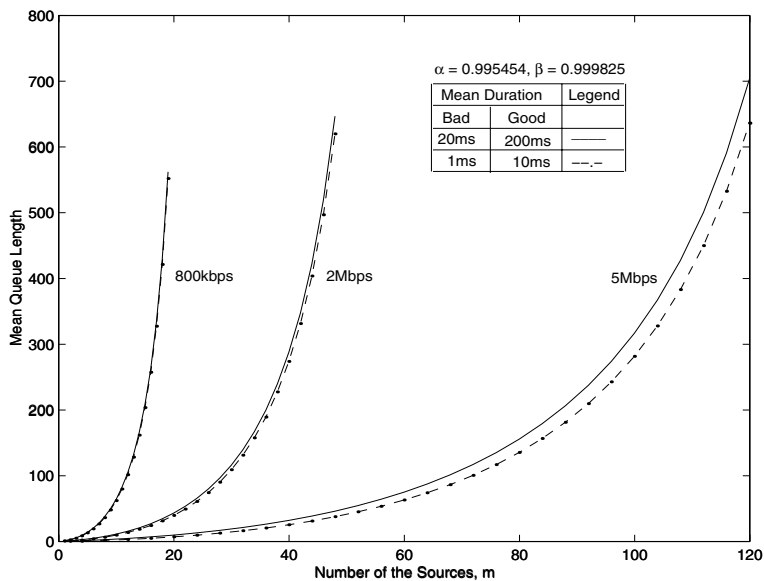


Fig. 4. Steady-state mean of the queue length versus the number of sources, m , for constant ratio of mean good to mean bad duration and different channel transmission rates

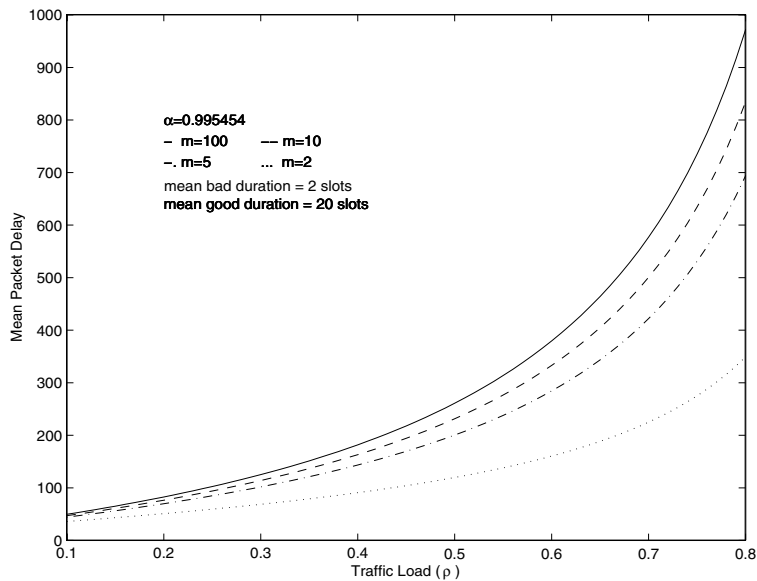


Fig. 5. Mean packet delay in number of slots versus traffic load with different number of sources, for a given channel error condition

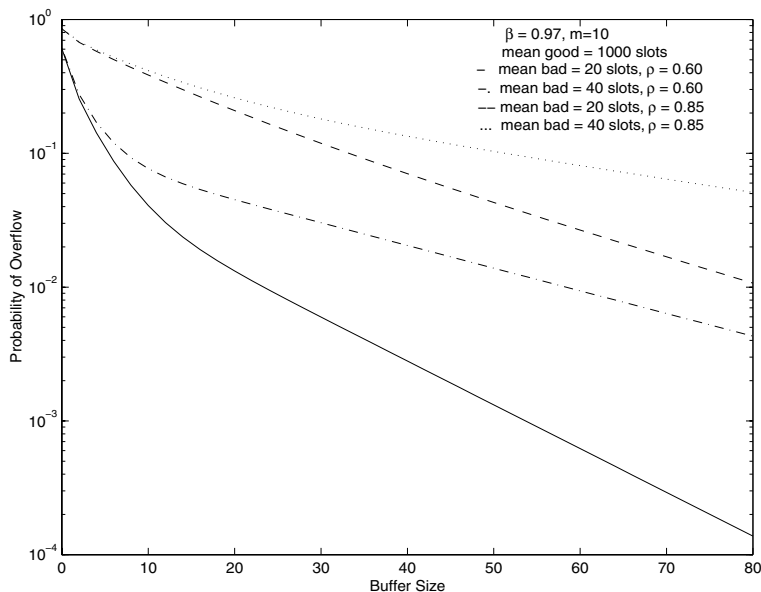


Fig. 6. Probability of buffer overflow versus buffer size

4 Conclusion

In this paper we have presented a discrete-time single server queuing analysis of a mobile user in a wireless ATM multiplexing system. The features of the model are a two-state Markov chain server and a correlated arrival process, consisting of the superposition of independent binary Markov sources. We determine the steady-state PGF of the queue length distribution as well as other performance measures.

References

1. D. Raychaudhuri and N. Wilson, "ATM-based transport architecture for multiservices wireless personal communication network," *IEEE JSAC*, vol.12, No.8, October 1994, pp.1401-1414.
2. E. N. Gilbert, "Capacity of burst-noise channel," *Bell Syst. Tech. J.*, vol.39, September 1960, pp.1253-1265.
3. E. B. Maglaris, et. al. "Performance Models of Statistical Multiplexing in Packet Video Communications", *IEEE Trans. Comm.* 36 (7), pp. 834-844, 1988.
4. D. Anick, D. Mitra and M. M. Sondhi, "Stochastic Theory of Data Handling Systems with Multiple of Sources," *Bell Syst. Tech. J.*, vol.61(8), 1982, pp.1871-1894.
5. San-qi Li, "A General Solution Technique for Discrete Queuing Analysis of Multi-Media Traffic on ATM," *IEEE Trans. on Communications*, vol.39, No.7, July 1991, pp.1115-1132.
6. H. Bruneel, "Queuing Behavior of Statistical Multiplexer with Correlated Inputs," *IEEE Transactions on Communications*, vol.36, No. 12, December 1988, pp.1339-1341.

7. M. Mehmet Ali and F. Kamoun, "A Transient Discrete-Time Queuing Analysis of the ATM Multiplexer," *Performance Evaluation Journal*, vol.32, April 1998, pp.153-183.
8. H. Takagi, Queueing Analysis, A Foundation of Performance Evaluation Volume 3: Discrete-Time Systems.
9. H. Bruneel and B. G. Kim, "Discrete-Time Models for Communication Systems Including ATM ", Kluwer Academic Publishers, 1993.
10. D. Towsley, "The Analysis of a Statistical Multiplexer with Nonindependent Arrivals and Errors", *IEEE Trans. on Communications*, COM-28, pp. 65-72.
11. X. Zhang, "A Performance Analysis of Wireless ATM Multiplexing System", M. Sc. Thesis, Department of Electrical and Computer Engineering, Concordia University, Montreal, Que., Canada, 2000.

A QoS Based Distributed Method for Resource Allocation in Unlicensed Wireless ATM Systems

Giannis F. Marias and Lazaros Merakos

Department of Informatics, University of Athens, TK15784,
{marias@mm.di.uoa.gr, merakos@di.uoa.gr}

Abstract. A decentralized multiple access technique for radio resource allocation for Base Stations (BSs) is presented. The proposed Distributed Dynamic Channel Reservation (DDCR) method applies to multicarrier wireless dynamic TDMA LANs, where self-organized BSs act as hubs offering wireless access to Mobile Terminals (MTs). The method is suitable for unlicensed wireless environment, and especially in scenarios where BSs of different providers are installed in a common coverage area. In the proposed approach, BSs are competing to access and reserve time on separate frequencies, using a dynamic TDMA/TDD technique. The paper introduces etiquette rules, and contention disciplines, whilst eliminates any hidden terminal phenomena. Additionally, it defines carrier selection rules, allowing BSs to select the less congested carrier, based on real time measurements. Finally, the paper evaluates and compares the performance of the contention disciplines and MAC techniques, through simulations.

1 Introduction

Wireless and mobile ATM (wmATM) as an emerging technology, is expected to enhance the traditional services offered by the existing cellular and wireless systems. Current implementations of wmATM utilize the 5GHz spectrum [1]. ETSI is currently developing standards for Broadband Radio Access Networks, which include the HIPERLAN type 2 system. This short-range variant is intended for private use as a wireless LAN. It will offer high-speed access (25 Mbit/s) to a variety of systems, including ATM. For HIPERLAN type 2, spectrum has been allocated in the 5 GHz range. Furthermore, the FCC has opened a 300 MHz unlicensed band at the 5GHz spectrum [2]. The basic feature of this band is that no company can monopolize a portion of it. It would therefore appear sensible to consider the use of the 5GHz band for unlicensed medium and short-range wmATM networks. On the other hand, the electromagnetic spectrum is limited. Thus, efficient methods for the allocation of the available frequency to mobile users and geographical cells remain critical. This paper introduces a QoS based variation of the Distributed Dynamic Channel Reservation (DDCR) method [3], [4]. According to the new method the BSs compete and reserve wireless resources in a multicarrier wmATM network, based on their QoS demands.

2 System Assumptions

wmATM system may be structured or unstructured. In the former case, all BSs can communicate via terrestrial links and they can use a protocol to regulate their access to the common wireless resources. In unstructured wmATM systems, the BSs cannot use a protocol to coordinate their access to the wireless resources. In such a case, all the BSs should follow a set of rules to compete for the radio resources. BSs following etiquette rules are referred to as self organized BSs. An example of an unstructured wmATM system is illustrated on Fig. 1.

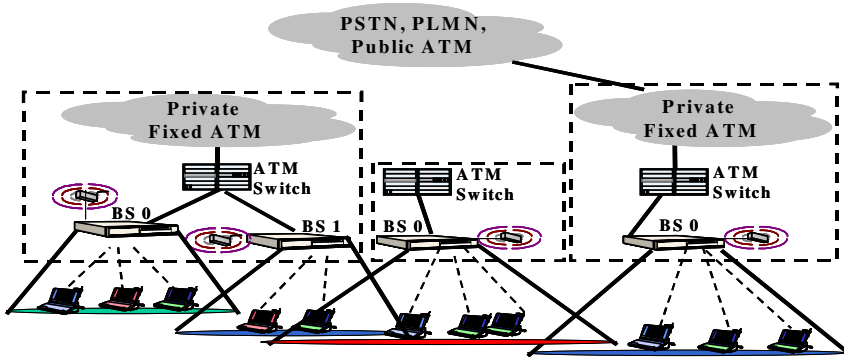


Fig. 1. A wmATM unstructured system

According to Fig. 1, different wmATM providers may install their wmATM networks in a common coverage area. In this high interfering environment, the installed BSs should first regulate their access to the shared wireless resources, and then spread the reserved resources to MTs. This paper assumes an unstructured system, which consists of multicarrier wmATM networks, where the self-organized BSs offering ATM wireless access to the MTs. For each wmATM network the transport architecture is based on a MDR mechanism, which uses a dynamic TDMA/TDD MAC scheme, as proposed in [5]. Other variations of dynamic TDMA approach can be assumed as well [6], [7], and [8]. Each MT maintains an association with one of the BSs, until it performs a handover. Each BS offers ATM wireless access to the MTs that are associated with it, and includes a MAC protocol to share the resources among its associated MTs. We assume multicarrier wmATM networks, where the available spectrum is divided into M carriers. We consider N BSs in the system, each of which competes for and reserves one of the M carriers at a time (i.e., single transmitter assumption). All the BSs and MTs are slot synchronized, and they use the same slot length. Each self-organized BS of a wmATM network uses a dynamic TDMA/TDD method to schedule downlink (BSs to MTs), and uplink (MTs to BSs) connections' data, during its reservation on one carrier. Thus, when BSs reserve a carrier, they exchange information with their associated MTs based on a single carrier scheduling protocol, such as MDR or MASCARA [7], which attempts to optimize the transmission in one carrier. On the other hand, prior to the reservation of a single carrier, the BSs follow the DDCR multicarrier scheduling method, in order to select the most befitting carrier (in terms of traffic load). Thus, DDCR attempts to schedule efficiently the traffic load, by treating all the slots on all carriers as a two-dimensional

scheduling problem (time and carrier). We assume that all the BSs use the same transmitted power level e.g., P_{BS} W, and all the MTs use the same transmitted power level e.g., P_{MT} W. Normally $P_{MT} \ll P_{BS}$. For instance Tx power can be 100mW (20dB) for small indoor coverage areas, or 1000mW for outdoor larger coverage areas (i.e., HIPERLAN type 2 and U-NII middle band). To sense idle carriers, a threshold of P_{th} dBm (e.g., -100 dBm) is adopted for the BSs. To cope with Turn Around Times (TAT, that is time required to switch from receive to transmit mode and vice versa) we assume that all MTs require one slot. When BSs communicate with MTs (i.e., during their reservations) they use one slot for TAT; otherwise BSs TAT is considered smaller (e.g., during competition period). Likewise, for the Switch Carrier Time (SCT, that is time required to switch carrier) we assume that all MTs require one slot. When BSs communicate with MTs they use one slot for SCT; otherwise BSs SCT is considered smaller (e.g., during competition periods).

3 Etiquette Rules of DDCR

The DDCR scheduling algorithm is presented on [3] and [4], and only some details of this algorithm are illustrated in this paper. To avoid congestion and interference conditions, DDCR separates control and data channels. Control channels are used to resolve contentions and to broadcast carrier status information. DDCR uses four special signal bursts. Burst signal is energy transmitted by BSs. They used in order to indicate certain conditions and to broadcast control information. Normally they use higher transmitting power levels than normal bursts (e.g., data slots on MAC frame), i.e., if BP_{BS} W is radiated for signal bursts, then $BP_{BS} > P_{BS}$. The signal bursts are:

- A BS transmits the Priority Burst Signal (PBS) during priority resolution phase, declaring its QoS demand.
- A BS transmits the Request Burst Signal (RBS), during the competition phase, declaring information such as perceived delay, and reservation period request.
- A BS transmits the End Burst Signal (EBS) at the end of its reservation on a carrier.
- A BS that reserves a carrier transmits the Utilisation Burst Signal (UBS).

3.1 DDCR Channels

According to DDCR process, once one or more interfering BS sense idle carrier the DDCR superframe start. This superframe consists of several channels (control and data), allowing BSs to solve the competition, to exchange data with their associated MTs, to release the carrier, and to broadcast control information. The DDCR channels are: a) the Priority Resolution Channel, b) the Contention Resolution Channel, c) the MAC Channel, d) the EBS Channel, and, e) the UBS Channel.

Priority Resolution Channel (PR-CH). Copying with ATM QoS, each BS estimates its Reservation Priority (RP). In DDCR, each BS competes with interferes in order to reserve a carrier for time period equal to its TDMA frame time. Thus, prior to the PR-CH channel, assume that a BS_i serves K_i ATM connections, classified as:

- $\{C_1, C_2, \dots, C_{k_i}\}$ real time connections (CBR and rtVBR)
- $\{V_1, V_2, \dots, V_{k_i}\}$ non real time connections (nrtVBR)

where, $k_x + k_y = K_i$. Real time connections (rtC) impose Cell Transfer Delay (CTD), whilst non real time connections (nrtC) impose Cell Loss Ratio (CLR) requirements. Each rtC C_i ($0 < i < K_x + 1$) insert a transfer Delay violation threshold, D_i^{thr} . Each nrtC V_i ($0 < i < K_y + 1$) insert a cell Loss violation threshold, L_i^{thr} . The RP for the BS_i is [9]:

$$RP_i = \frac{1}{2} \frac{\sum_{i=1}^{k_x} \frac{D_i}{D_i^{thr}}}{k_x} + \frac{1}{2} \frac{\sum_{i=1}^{k_y} \frac{L_i}{L_i^{thr}}}{k_y}$$

The PR-CH period consists of a constant number of slots. This period is further divided to PR-CH minislots (p-slots). Each p-slot order corresponds to a particular RP. For instance assuming a 5 p-slot granularity, the 1st p-slot of the PR-CH period corresponds to $RP \leq 0.2$, the 2nd p-slot corresponds to $0.2 < RP \leq 0.4$, and the last p-slot corresponds to $0.8 < RP \leq 1$, as shown in Fig. 2. According to the estimated RP_i, the BS_i will broadcast its PBS_i during the corresponding p-slot. If D_{PBS} is the duration of signal PBS, and D_{ps} is the duration of p-slot, then $D_{ps} > D_{PBS}$, and $TAT < D_{ps} - D_{PBS}$. This allows BSs to switch from transmit to receive mode and sense PBS broadcast on the next order p-slot. A backlogged BS, i.e., with low PR, sense the PBS burst of the BS illustrating higher PR, because the latter will broadcast its PBS using a higher order p-slot. Backlogged BSs select a new carrier to compete for it.

Contention Resolution Channel (CR-CH). On the PR-CH channel we have adopted a dimensioning scheme (i.e., number of p-slots) to represent RPs with a certain regularity. Thus, it is possible for two more BS to use p-slots of the same order to broadcast their PBSs, even if their RPs have different values (e.g., on the 2nd decimal digit of RPs). To overcome this problem we introduce the CR-CH.

During CR-CH period each BS, survived from priority resolution phase, broadcast its reservation requests (through the RBS), and realizes the reservation requests of other interfering BSs. Reservation requests represent either current MAC frame time length, or mean reservation delay, or both. The CR-CH comprises of an integer, but not fixed, number of slots, each of which is divided to a fixed number of minislots (c-slots), as Fig. 2 shows. The RBS signals are transmitted on continuous c-slots. We introduce a granularity factor g , $0 < g < 1$. If T is the reservation request (MAC length, delay, or both) in slots, then RBS will use $[g * T]$ c-slots for its transmission. If D_{cs} is the duration of c-slot, then $TAT < D_{cs}$. This allows BS to switch from transmit to receive mode and sense RBS broadcast by another BS.

Longest Job First (LJF) discipline

According to this discipline the winner of the competition is the BS with the larger reservation request (i.e., having the larger MAC frame). Thus, if TFD_i is the number of slots the BS_i wishes to reserve on this carrier (i.e., current time frame length), then the BS_i transmits a RBS_i of $[g * (TFD_i)]$ c-slots. The winner (survivor) BS is the one that broadcasts the larger RBS

Delayed Job First (DJF) discipline

The winner of the competition is the BS that received the highest mean delay from its previous reservation attempts on any carrier. Each BS_i records the last reserved slot in any carrier, say T_{Ri} , and switches to a carrier in order to compete for it. Then it

calculates the $T_M = \text{mean}(T_{R_i})$. If the BS_i is involved in the competition, BS_i transmits a RBS_i signal, equal to $\lceil g^*(T_M) \rceil$ c-slots. The mechanism is identical for the LJF and DJF disciplines. In the former case the RBS is proportional to the frame size, whilst in the latter case the RBS is proportional to the received reservation delay.

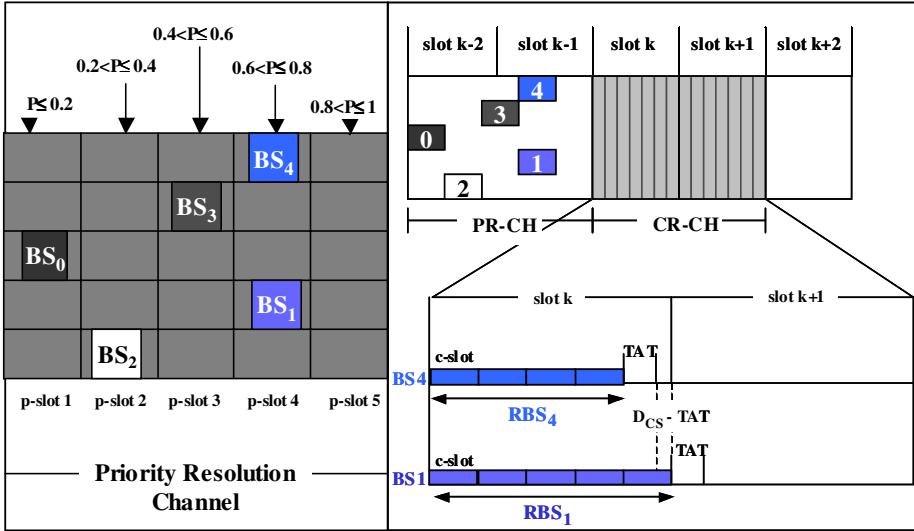


Fig. 2. BSs broadcast their reservation priorities through PBS signal bursts on the corresponding p-slot of PR-CH (left figure). The BSs that will survive are those broadcasting on the higher order p-slots, in our case during the 4th p-slot. These BSs will broadcast their RBS signals during CR-CH (right figure). The winner is the BS that broadcasts longer RBS.

Delayed and Longest Job First (DLJF)

This is a combination of LJF and DLF disciplines. The winner of the competition is the BS that experiences the higher reservation delay, and requests the larger time frame. Thus, if T_{R_i} is the last reserved slot of a BS_i in any carrier, and the contention for a carrier involves the BS_i , and TFD_i is the number of slots the BS_i wishes to reserve on this carrier (i.e., current time frame length), then the BS_i transmits a RBS_i of $g^*(TFD_i + T_M)$ c-slots, where $T_M = \text{mean}(T_{R_i})$. A backlogged BS sense the RBS burst of the BS illustrating higher reservation request, because the latter will broadcast an RBS using at least on more c-slot. Backlogged BSs select a new carrier, among the M candidates, to compete for it. The survivor is the BS that has completed its RBS transmission, switched on receive mode and sense the carrier idle.

Medium Access Control Channel (MAC-CH). This period is used for data transfer, i.e., accommodates the MAC frame. It consist of

- Frame Header Broadcast Channel (FHB-CH), Within this channel the BS broadcast its MAC frame slot map to its associated MTs, i.e., in which slot each MT can send or receive data, and when downlink, uplink and MTs contention periods start..
- Down Link Data Channel (DLD-CH), with variable duration, accommodating information sent from BS to MTs.

- TAT Channel (T-CH), which occupies one time slot and allows MTs or BSs to switch from receive to transmit mode and vice versa.
- Up Link Data Channel (ULD-CH), with variable duration, accommodating information sent from MTs to BS.
- MTs Contention Channel (MC-CH) which allows associated MTs, with no allocated ULD-CH slots, to request reservation slots, or accommodates association requests from MTs.
- Frame Trailer Channel (FT-CH), which occupies one slot. Within this channel the BS broadcast the FT to the associated MTs. FT includes a visiting list of the carriers that the BS will visit sequentially until a successful reservation.

End Burst Signal Channel (EBS-CH). This channel uses one slot, and it is used for the broadcast of the EBS signal.

Utilisation Burst Signal Channel (UBS-CH). This channel uses one slot, and during this period the UBS signals are broadcast. This signal denotes the number of BSs that compete or use the carrier during a recent period. Each BS_i for each carrier F_r , continuously updates the value of $UBS_{i,r}$. The notation $UBS_{i,r}$ denotes the UBS maintained by the BS_i for carrier F_r ($0 \leq r < M$). $UBS_{i,r}$ is updated according the rules presented on [3] and [4]. A BS, which has reserved a carrier, broadcasts a UBS, after a predefined period of R slots. All the BSs know that the UBS signals are broadcast every R slots (UBS-CH). If BS_i during the UBS-CH uses a carrier F_r , it broadcasts the $UBS_{i,r}$ for this carrier, otherwise it receives the transmitted UBS.

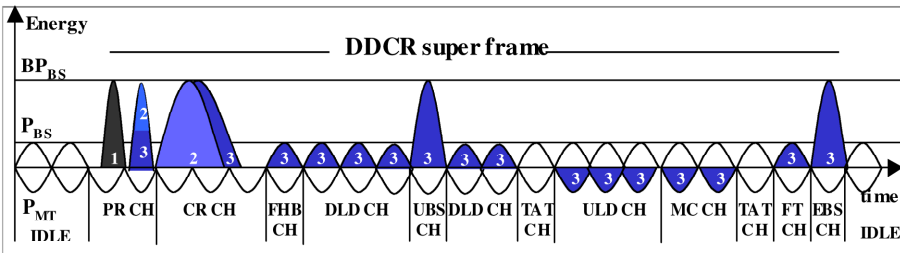


Fig. 3 DDCR superframe and its Channels for 3 interfering BSs and one available carrier. During PR-CH the BSs 2 and 3 use the same order p -slot to transmit their PBS, whilst BS 1 uses lower order p -slot. The BS 1 losses the competition, whilst the BSs 2 and 3 are allowed to broadcast their RBS during CR-CH. The BS 3 broadcast longer RBS, and reserves the carrier.

3.2 DDCR Process Steps

When choosing a carrier, the BS should choose the carrier illustrating the less congestion. A BS_i keeps its Selection Parameters (SP_i) list, as follows: $SP_{i,r} = (CurrentTime_i - LastVisitTime_{i,r} + 1) / UBS$. A BS_i selects the carrier illustrating the minimum $SP_{i,r}$ value. More details are discussed in [3] and [4]. A BS_i follows the following steps in order to select, compete, reserve and release a carrier.

1. The BS_i forms the SP list and select the $\min\{SP_{i,r}\}$, say c , i.e. selects the carrier F_c .
2. The BS_i switches to carrier F_c , listens to the F_c and if it is reserved by other BS(s) then it waits until it will recognises an EBS, otherwise it goes to step 3.

3. If the BS_i receives an EBS transmitted by other BS, or if no other BS_j , $j \neq i$, uses the carrier F_c for two consecutive slots, the competition period starts.
4. The BS_i , based on the RP_i , transmits its PBS to the matching p-slot of the PR-CH.
5. On the completion of the PBS transmission, the BS_i returns to the listening mode, on carrier F_c . If the BS_i detects the F_c busy (due to transmitted PBSs by other competing BSs on higher order p-slots), the BS_i loses the competition for carrier F_c . The BS_i should change carrier (i.e., uses the next carrier F_n of the visiting list) and goes to step 2 (assume that $F_c = F_n$). Otherwise it goes to step 6.
6. The BS_i estimates its reservation request, and broadcasts its RBS during the CR-CH (according to discipline LJF, DJF or DLJF).
7. On the completion of the RBS transmission, the BS_i returns to the listening mode, on carrier F_c . If the BS_i detects the F_c busy, the BS_i loses the competition for carrier F_c . The BS_i should change carrier (i.e., uses the next carrier F_n of the visiting list) and goes to step 2 (assume that $F_c = F_n$). Otherwise it goes to step 8.
8. The BS_i , as a winner, reserves the carrier, and exchanges information with its associated MTs during MAC-CH.
9. The BS_i during FT-CH uses the $UBS_{i,k}$ values ($k < M$), calculates the SP elements, produces the carrier visiting list and broadcast this list within FT.
10. On the completion of FT broadcasting it transmits the EBS, and goes to step 1.

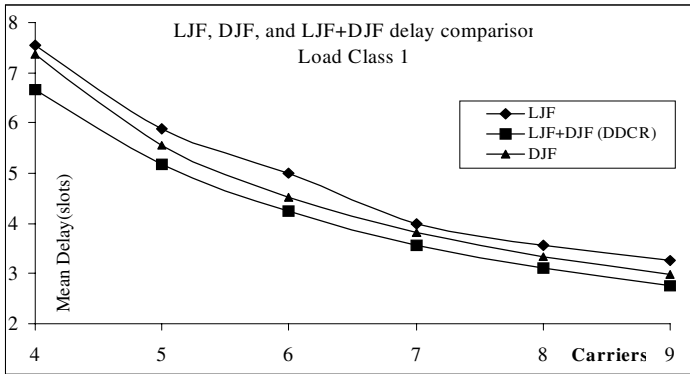


Fig. 4. Comparison of the DLJF (LJF+DJF), LJF, and DJF disciplines for load class 1, for $N=10$ BSs, and varying number of available carriers.

4 Simulation Environment and Results

To evaluate the DDCR performance, simulations were performed using the OPNET Simulator. In the simulations, each dynamic TDMA frame was assumed to contain a fixed number of 5 slots for FH, FT, TAT, and MC channels. The channel speed was set to 20Mbps/sec. Each slot was of 54 bytes long. For the simulations, we used a combination of both CBR and VBR connections. For each CBR connection, a simple periodic ATM cell generator was used. We assumed 64Kbps CBR connections. We used 50 CBR sources (25 uplink and 25 downlink) per BS. On the other hand, each VBR source is modeled by a Discrete time Batch Markov Arrival Process. We consider the slot as the time unit. According to [10], the traffic load produced by a

VBR source can be approximated by the super-position of U equivalent ON/OFF minisources. For each VBR source, μ was set to 256Kbps and σ^2 was set to 128Kbps, where μ and σ^2 are the mean and the variance of the transmission rate, respectively. We considered two different load classes of VBR connections. For class 1 and 2, we used 20 VBR and 30 VBR connections per BS, respectively.. From the simulations is concluded that the delay illustrated when RR, or RC methods were used is higher than the delay obtained when the DDCR method was used. More particularly, for load class 1, the DDCR method achieves more than 10% and 30% delay improvement, when compared with the RC and RR disciplines, respectively. For load class 2, the DDCR delay improvement varies from 15% to 35%, when compared with the RC discipline, and from 50% to 80%, when compared with RR discipline. Fig. 4 illustrates the delay obtained when the combination of the LJF and the DJF disciplines is used when only the LJF is used, and when only the DJF is used. From Fig. 4 we observe that the combination of the LJF and DJF causes smaller reservation delay. This observation was confirmed for load class 2, and for all the combinations of M and N, as well. Fig. 5 illustrates the fluctuation of the delay between two consecutive reservations, for 10 BSs, 4 available carriers, and for load 1 class. From Fig. 5 we conclude that the mean delay decreases with time. The final value of the mean delay, averaged after 1500000 slots, is about 6.7 slots, and that the delay fluctuation is attenuated smoothly with time, and approximates the final mean value after 25000 slots of simulation. Thus, it is concluded that the BSs maintain a realistic view of the congestion on carriers after the first 25000 slots, for this simulation environment

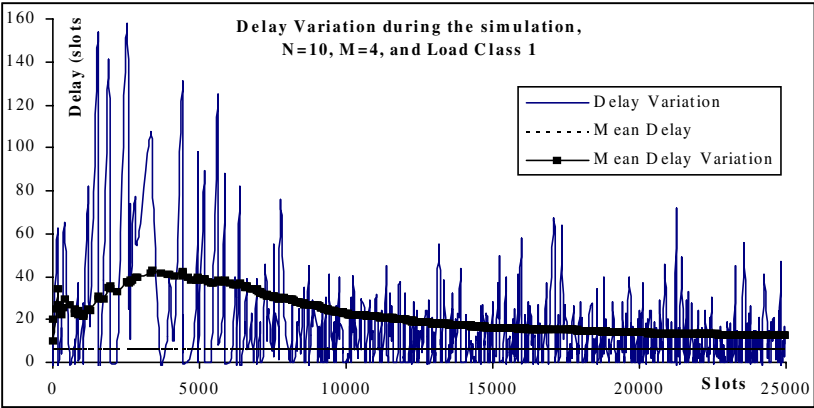


Fig. 5. Delay variation for load class 1, for 10 BSs, and 4 available carriers.

Fig. 6 shows the fairness of the DDCR method. According to this figure, for a wmATM system consisting of 10 BSs, for M=6, and for load class 1, for DLJF discipline, the BSs with identical traffic demands and equal number of competitors will experience similar reservation delay. Let NC_K the set of BSs having K competitors, i.e., interferers. Then, for this installation, $NC_6=\{BS_0, BS_1, BS_5\}$, $NC_5=\{BS_2, BS_7\}$, $NC_4=\{BS_3, BS_4, BS_6, BS_8\}$, and $NC_3=\{BS_9\}$. In Fig. 6 the notation $BS_i(k)$ denotes the number k of competitors of a BS_i .

5 Concluding Remarks

We have introduced QoS based competition rules for a distributed carrier reservation method, which applies to interfering BSs in an unlicensed multicarrier wireless ATM system. The DDCR mechanism is immune to topology changes (e.g., installation of new BS in a common area), does not increase power consumption on MTs, and requires no frequency preplanning. Furthermore, DDCR imposes no limit on the number of BSs operating in a common area. From the simulation results, we have observed that a combination of the LJF and DJF disciplines performs better in terms of reservation delay. Furthermore, the simulation results show that when combining DDCR process with congestion estimation the reservation delay stabilizes. Thus, the DDCR process could be combined with a distributed Wireless Call Admission Control. The latter could take into account the DDCR decisions, determine if the system is under heavy load, and regulate the admission policy, accordingly.

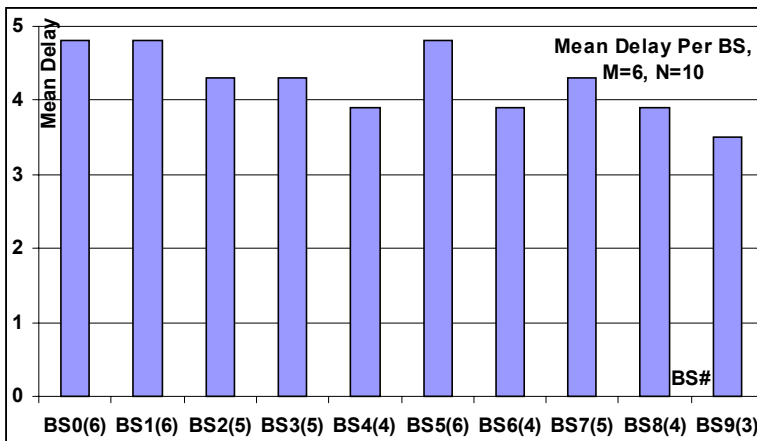


Fig. 6. DDCR achieves fairness on reservation delay.

References

1. ACTS Project, "The Magic WAND", Deliverable 1D5, "Wand System Specification", CEC Deliverable Number AC085/NMP/WDA/DS/P/038/b1, Aug. 1998.
2. FCC "Notice of Proposed Rule Making", ET Docket No. 96-102, Apr. 1996.
3. G. Marias, and Lazaros Merakos, "A distributed dynamic channel reservation for wireless ATM LANs", in Proc. IZS2000 Seminar on Broadband Communications, Zurich, Switzerland, Feb. 2000.
4. G. Marias, and L. Merakos, "A Distributed Multiple Access Technique For Radio Resource Allocation In Unlicensed Wireless ATM LANs", in Proc. PIMRC 2000, London, UK, Sept 2000.
5. D. Raychaudhuri, and N. Wilson, "ATM-based transport architecture for multiservices wireless PCN", IEEE JSAC, vol. 12, no. 8, Oct. 1992.
6. N. Wilson, D. Raychaudhuri, et. al., "CDMA vs Dynamic TDMA for access control in an integrated voice/data PCN", Proc 1st ICUPC, Sept. 1992

7. L. Merakos, et al., "A medium access control framework for wireless ATM networks", in Proc. IUMC, Thessaloniki, Greece, Sept. 1996.
8. N. Passas, L. Merakos, S. Paskalis, D. Vali, "Quality-of-Service-Oriented medium access control for wireless ATM networks", IEEE Comm. Mag. Nov. 1997
9. ACTS Project, "The Magic WAND", Deliverable 3D6, "Wand DCA Scheme," CEC Deliverable AC085/INT/ACT/DS/P/034/b1, Aug. 1998
10. C. Blondia, and O. Casals, "Performance analysis of statistical multiplexing of VBR sources", in Proc. INFOCOM '92.

An Adaptive Error Control Mechanism for Wireless ATM

Peter R. Denz and Arne A. Nilsson

North Carolina State University

Abstract. In order to provide a ubiquitous telecommunications network which merges the concepts of wireless mobile communications and the use of ATM as a transport mechanism, the problem of the high and variable bit error rate of the wireless network must be addressed. This paper provides an adaptive error control coding feedback mechanism to alleviate this problem in wireless ATM networks. The scheme combines convolutional coding, optimal interleaving, puncturing, and channel state estimation to achieve an appropriate error correction scheme. Periodic feedback is provided to the source machine indicating changes in the channel state and triggering changes in the coding scheme.

1. Introduction

ATM has the potential of becoming ubiquitous on all computer platforms. Thus to avoid the problem of protocol conversion, we must provide ATM for wireless systems, as these will grow more in popularity. This wireless version of ATM has many problems which must be overcome before it can be a reality. The ATM protocol was designed with the assumption of a low-noise fiber physical medium and a fixed addressing structure. In a wireless network, the physical link is very noisy and the users are potentially mobile. Not only is the physical channel very noisy, but the noise level varies according to the current physical environment. In this paper we propose a dynamic error control coding mechanism for wireless ATM to lessen the detrimental effects of the high and variable noise which exists on this channel.

The ATM architecture was designed to function over a low noise fiber channel, thus relaxing the need for error control inside of the network. An implementation of ATM over a noise-varying wireless channel, on the other hand, requires additional error control mechanisms to be in place.

In general, error control can be of two forms: Automatic Repeat reQuest (ARQ), which involves error detection and retransmission of erroneous messages, and Forward Error Correction (FEC), in which redundancy is added to the data to enable correction of some error pattern at the receiver. We propose the addition of FEC, in the form of convolutional coding, such that the high bit error rate of the wireless channel is lessened. Furthermore, we have developed an adaptive FEC mechanism such that more redundancy is provided in the face of a very noisy channel and less redundancy is provided in the face of a less noisy channel.

In this paper, we discuss various techniques which combine to form our adaptive error control coding mechanism. Section 2 discusses the Gilbert channel model and

parameter estimation, section 3 introduces the interleaving concept, section 4 discusses puncturing which will be used to dynamically change the coding rate, section 5 overviews the coding model, and sections 6 and 7 present the adaptive error control coding mechanism and simulation results.

2. Gilbert Channel Model

We model the noisy wireless channel using the Gilbert channel model, since it models burst noise well. This channel model consists of a two state Markov chain. These states represent the *good* state, in which no error may occur, and the *bad* state, in which errors can occur with probability h . The state transition probabilities are shown in the diagram of figure 1. The expected length of the stay, in bits, in the *good* and *bad* state, given by the geometric distribution, is $\frac{f_i}{c_i - f_i}$ and $\frac{1}{1 - \alpha}$, respectively [8].

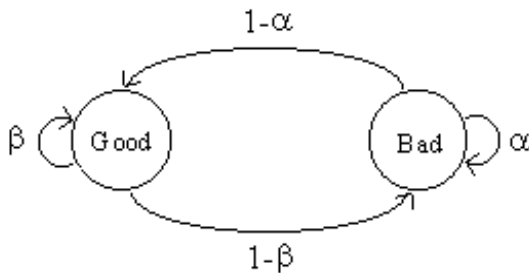


Fig. 1. Gilbert Channel Model

Given a sufficiently long error vector, we can calculate estimators $\hat{\alpha}$, $\hat{\beta}$, and $\hat{h}$ for the current channel state as follows. Based on the error vector, calculate $a = P[1]$, $b = P[1 \mid 1]$, and $c = P[1 \mid \text{there exists a directly preceding and a directly succeeding error}]$. Finally, the estimators can be calculated as follows:

$$\hat{\alpha} = \frac{ac - b^2}{2ac - b(a + c)}$$

$$\hat{h} = 1 - \frac{b}{q}$$

$$\hat{\beta} = 1 - \frac{ap}{1 - h - a}$$

The derivation of these equations appears in [4].

3. Interleaving

The convolutional encoder and Viterbi decoder pair perform best when errors are spread apart through the bit stream, providing what is called a guard space between bad bits. Unfortunately bit errors on the physical medium tend to be correlated and occur in bursts. This severely impacts the performance of the error correction coding system. To alleviate this problem, we spread the error bursts apart using a mechanism called an interleaver. The interleaver is simply a square matrix into which the bit stream is read in row-wise and read out column-wise. The uninterleaving mechanism performs the same function to properly reorder the bits at the destination. This interleaving is applied after the convolutional encoder at the sender and prior to the Viterbi decoder at the receiver.

The size of the interleaver matrix is an important parameter which we would like to choose carefully to yield optimal interleaver performance. Through simulation we have discovered the seemingly intuitive fact that the interleaver performance improvement is optimal when the interleaver matrix's side is equal to the expected bad burst length. This provides the maximum guard space between bad bits.

Too small of an interleaver will cause bad bits to appear in pairs on the output of the interleaver since the bad burst continued to the next row of the matrix. Or, even worse, an extremely small interleaver may not even contain a proper mix of good and bad bits. Too large of an interleaver also may lead to non-optimal performance since multiple bad bursts may appear inside of the matrix. Also, if the matrix is excessively large, then the tail end of the message may not fill out the entire message, thus requiring a significant overhead in padding bits which will have to be transmitted.

4. Puncturing

Puncturing is a technique in which bits are systematically removed from the output of the convolutional encoder according to a pattern specified by the puncturing matrix.

For instance, the matrix $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ specifies that every sixth bit is to be deleted. At

the receiver, the same puncturing matrix is known, and, prior to the Viterbi decoder, random bits are entered into the bit stream at the same locations where the bits were removed. This way, we are artificially adding errors to the bit stream and allowing the Viterbi decoder to recover from these errors. This yields a reduction in the amount of data which will be sent over the transmission channel.

This puncturing technique allows us to modify the rate of the convolutional code while maintaining the same encoder/decoder pair. We will use puncturing to allow the error correction mechanism to adapt to current channel conditions [9].

5. Coding Model

We have developed a simulator based upon the model shown in figure 2. The simulator generates packets and encodes these packets using a (2, 1, 5) convolutional encoder. After encoding, the packet bit streams are punctured and interleaved according to the current coding scheme. These resulting bit streams are passed into a bursty channel simulator where errors are introduced to the data. Next, the streams are un-interleaved and un-punctured. Finally, the erroneous bit streams enter a Viterbi decoder which attempts to correct all the errors in the bit streams.

If the decoder can correct all of the errors introduced into the message, then we construct the error vector representing the noise pattern seen on the channel (discussed in section VI), calculate estimators for the current channel state and dynamically change the coding scheme according to these results. If the decoder is unable to correct all of the errors in the packet, a retransmission is performed.

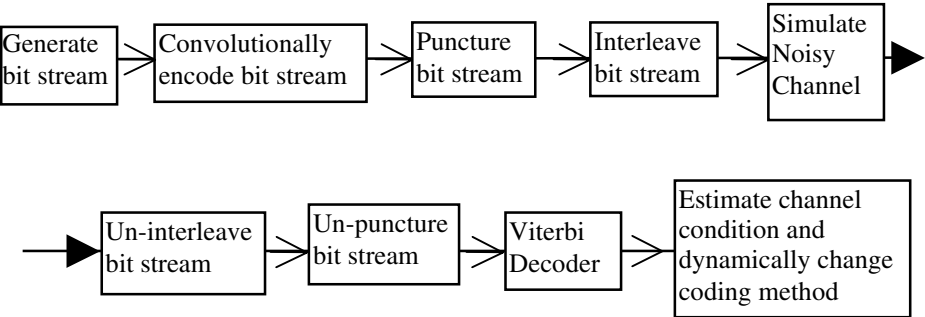


Fig. 2. Simulator Block Diagram

6. Adaptive Error Control Coding Mechanism

We would like to incorporate the aforementioned techniques into a dynamic error control coding mechanism which adapts to the long-term changing noise conditions of the wireless channel. In this mechanism we will sample the current channel noise levels at the destination, adjust the coding mechanism, if needed, and send a control packet back to the sending machine to communicate the adjustment.

Sampling the current noise level on the channel is done at the destination machine upon receipt of a packet. The receiver attempts recovery of the original message by reversing the encoding sequence. If recovery is successful, which can be checked simply through a CRC calculation, then an error vector can be generated as follows; otherwise one or more uncorrectable errors have occurred and retransmission is necessary.

The receiver can take the corrected message (k) and re-encode it using the same encoding sequence that had original been used to yield j of figure 3. Now we calculate j XOR r to yield the desired error vector. Finally, we perform Gilbert model

parameter estimation on this error vector to obtain $\hat{\alpha}$ and $\hat{\beta}$. Now we have measurements of the expected *good* and *bad* burst lengths.

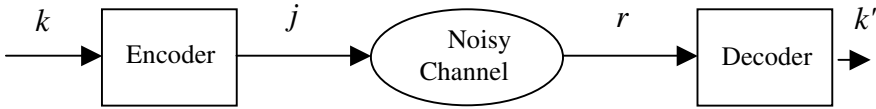


Fig. 3. Message Encoding/Decoding

At each message receipt, we can calculate the expected good and bad burst lengths and compare these to their respective values of the last code adjustment point. If the percent deviation of either of these is greater than a certain threshold, then we may initiate a new coding method to account for the changed noise level.

At each code adjustment point we potentially select a new optimal interleaver strategy and a new puncturing level. The interleaver selection comes directly from

the channel noise estimate $\hat{\alpha}$, which tells us that the expected noise burst is $\frac{1}{1 - \hat{\alpha}}$, and thus we use this as the next interleaver matrix.

Now we must choose an appropriate puncturing mask according to the current channel conditions. We have an array of available puncturing matrices to choose from varying from a very aggressive mask which deletes one out of every four bits up to a mask which deletes no bits (i.e., no puncturing) and many in between these extremes. We perform a binary search over these available matrices, choosing a new one at each code adjustment point, until we find the most aggressive mask which does not yield uncorrectable errors. We do this by becoming more and more aggressive until we see our first uncorrectable error at which point we back off the matrix to the previous one and maintain this one until either the noise level changes or another uncorrectable error is encountered.

7. Simulation Results

To test this adaptive error control coding mechanism through simulation, we have constructed a noisy environment in which the noise levels (i.e., the α and β parameters) change periodically as illustrated in figure 4. This noise pattern is used as the basis for all of the following simulated runs and yields the raw channel bit error rate (before any error correction) shown in figure 5.

The graph shown in figure 6 compares the performance of the dynamic coding mechanism to the various cases of static coding (using various puncturing matrices), in terms of the probability of frame retransmission. Of course the case of static coding with no puncturing has a lower probability of frame retransmission than our new dynamic coding mechanism. However this case also puts many more bits out onto the line. Looking at the other curves, "Static; Punct = 1," which corresponds to the most

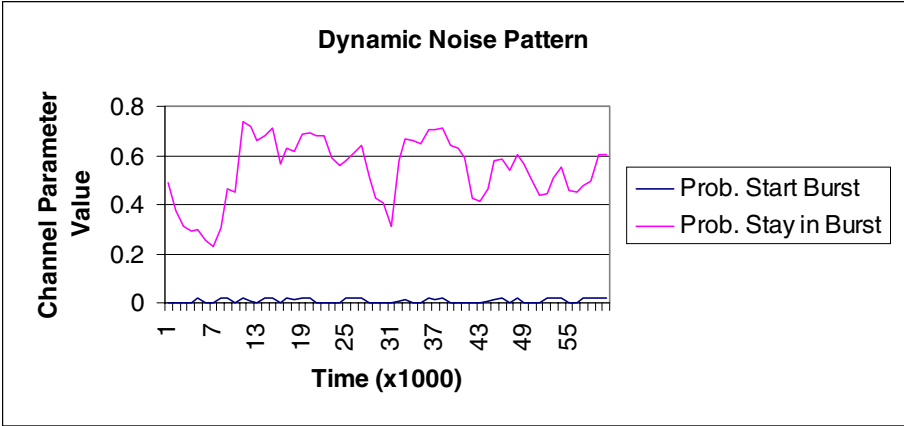


Fig. 4. Dynamic Noise Pattern

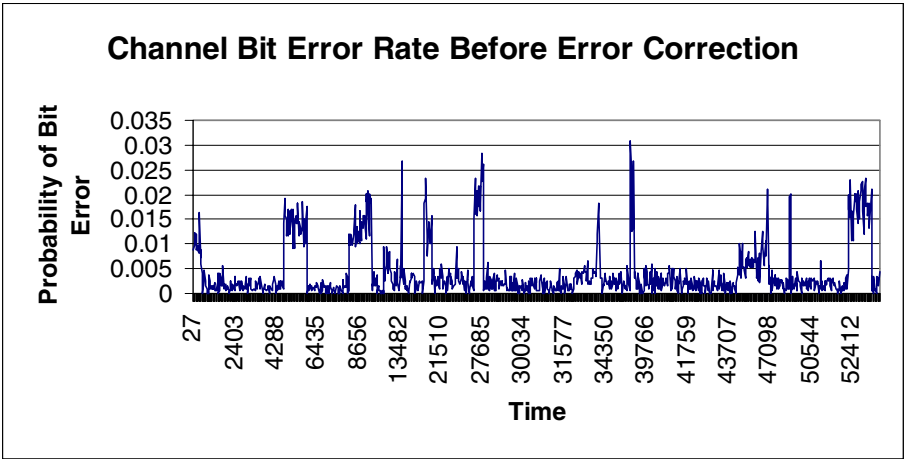


Fig. 5. Channel Bit Error Rate Before Error Correction

aggressive puncturing matrix $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, is too aggressive in bit removal, as the probability of frame retransmission shoots up to 1.0. The remaining data series; "Static; Punct = 2," "Static; Punct = 3," and "Static; Punct = 4"; represent the progressively weakening puncturing matrices, $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$, and $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$ respectively. As we see the curves for the progressively weaker coding schemes, the probability of frame retransmission drops, as we expect, but remains worse than the dynamic coding case.

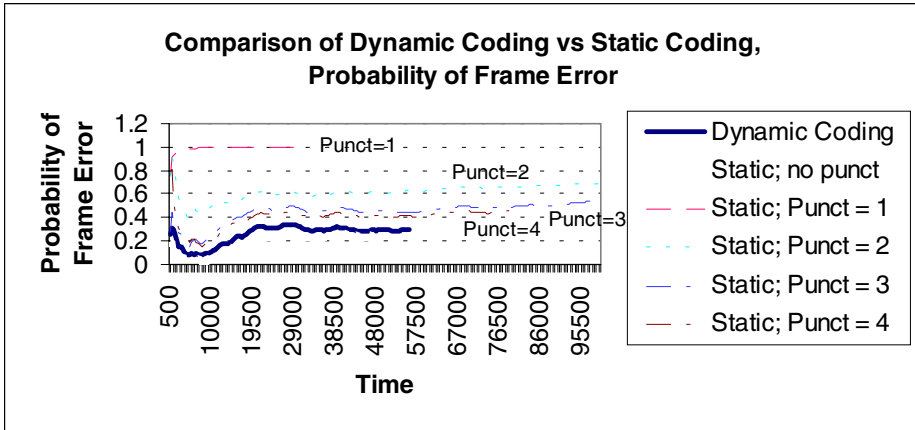


Fig. 6. Probability of Frame Retransmission: Dynamic vs. Static Coding

Some of these curves do not extend to the right edge of the graph. This is due to the fact that the simulation has ended. The simulations which entail aggressive puncturing matrices require more time due to all of the retransmissions that are necessary. The average percentage of bits on the line for the run for the dynamic coding method using this particular noise pattern is 96%, due to the high noise levels observed on this particular channel. Less noisy scenarios will, of course, yield a lower percentage of bits on the line.

Figure 7 is based upon the same set of simulation runs as figure 6. Now we are looking at a plot of the percentage of bits that actually get transmitted onto the line. As expected, all the static coding schemes have a constant percentage of bits going out onto the line, according to the puncturing matrix in use. However, the dynamic coding case varies over time as we constantly switch puncturing rates.

Figure 8 allows us to see the correlation between the channel noise levels and the performance of the dynamic coding scheme. Notice that during periods of high noise, a less aggressive matrix is used. However, during low noise periods more aggressive matrices are used.

In this comparison, however, one must look at both the values of the probability of starting a burst as well as the probability of remaining in a burst. Small changes in the former actually have stronger effects on the bit error rate than changes in the latter. This effect makes the dynamic noise graph deceiving in that the probability of remaining in a burst is visually more central in the graph and varies more, whereas the probability of starting a burst has a much smaller deviation range. To illustrate the difference in the effects of these two parameters, from the geometric distribution we see that varying the probability of starting a burst from 0.02 to 0.05 changes the expected *good* state stream length from 50 to 20. Likewise, varying the probability of remaining in a burst from 0.6 to 0.4 changes the expected *bad* burst length from 2.5 to 1.67. Notice here that a small change in the probability of starting a burst yields a very significant change in the noise level, whereas a much larger change to the probability of remaining in a burst yields a very small change in the noise level.

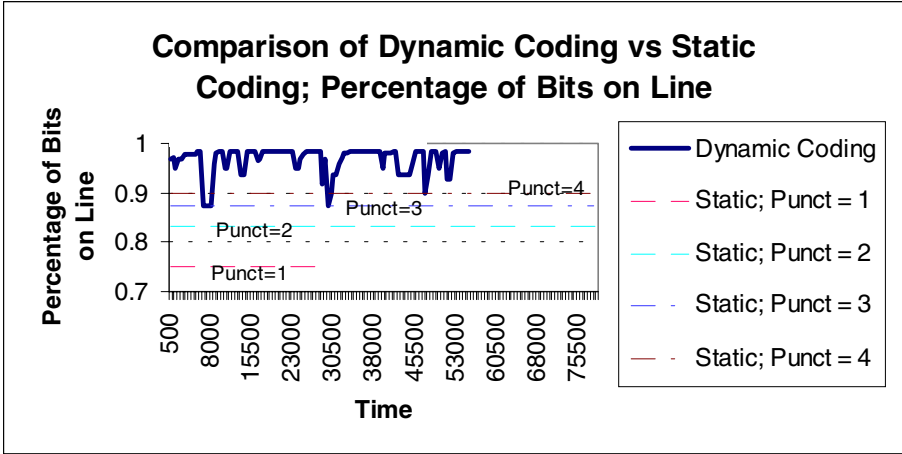


Fig. 7. Percentage of Bits on Line: Dynamic vs. Static Coding

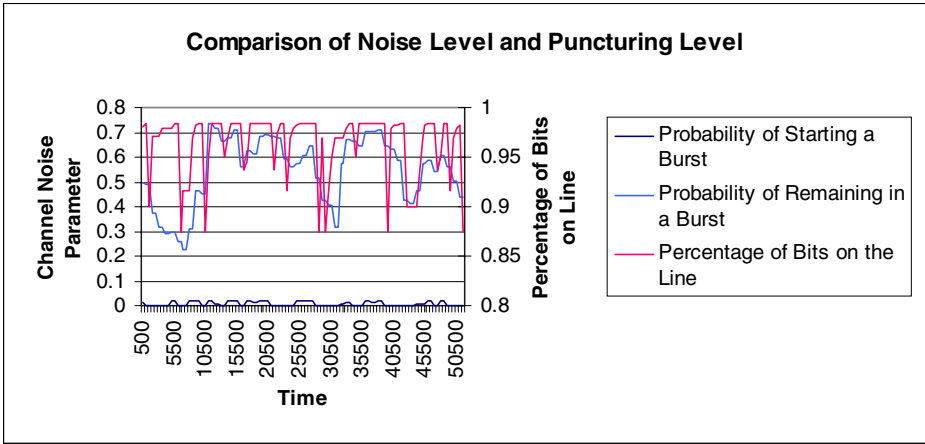


Fig. 8. Noise Level and Puncturing Matrix Correlation

8. Conclusion

In this paper, we have proposed an adaptive error correction technique which could be used in wireless ATM networks. A strong error correction code scheme will be essential to the functioning of a wireless ATM protocol due to the high bit error rate of the underlying wireless channel, however this scheme must also adapt itself to the changes in the current wireless channel conditions. We have seen the beneficial affects that interleaving has on the error correction capability by effectively spreading out burst errors and providing the necessary guard space for the Viterbi decoder.

Also, we have shown that the ideal interleaver matrix has a side equal to the expected error burst length.

Puncturing provides a convenient method by which the rate of the convolutional code can be changed without changing the actual encoder/decoder pair.

In order to provide an effective adaptive error control mechanism, we must somehow gauge the current noise level of the channel. Gilbert's formulas provided good estimates for the channel parameters. We use this parameter estimation technique, in conjunction with the optimal interleaver prediction scheme and puncturing to provide a dynamic coding mechanism which balances the probability of frame retransmission and the coding rate.

The dynamic error control coding mechanism developed in this paper is not restricted in any way to wireless ATM systems. Wireless ATM is just one current system under development which would benefit through its use. The mechanism would also benefit any other wireless networking system which suffers from a high and variable noisy environment.

Many hurdles will have to be overcome before wireless ATM can become a reality. Bit error rates will no longer be negligible and will require forward error correction coding to hide this increased bit error rate from the higher layers on the protocol stack. Users will no longer be fixed in one location, thus complicating the handling of ATM QoS guarantees. Security is an increasingly important issue that will need to be addressed for the new environment. Once these, and other problems are solved, computer networks will evolve in such a way that the underlying physical network will be hidden from the user; thus approaching our goal of ubiquitous and tetherless access to the computer network.

References

- [1] A.S. Acampora and M. Naghshineh. Control and Quality-of-Service Provisioning in High-Speed Microcellular Networks. *IEEE Personal Communications*, Second Quarter 1994, pp. 36-42.
- [2] T. Alanko, M. Kojo, H. Laamanen, M. Liljeberg, M. Moilanen, and K. Raatikainen. Measured Performance of Data Transmission Over Cellular Telephone Networks. *Computer Communication Review*, 24(4) pp. 24-44, October 1994.
- [3] H. Armbruster. The Flexibility of ATM: Supporting Future Multimedia and Mobile Communications. *IEEE Communications Magazine*, February 1995, pp. 76-84.
- [4] P.R. Denz. Performance of Error Control Coding Techniques and the Development of a Dynamic Error Control Coding Mechanism for Wireless ATM. Doctoral Dissertation, Department of Electrical and Computer Engineering, NC State University. 1999.
- [5] A. Dholakia, M.A. Vouk, and D.L. Bitzer. A Lost Cell Recovery Technique Using Convolutional Coding at the ATM Adaptation Layer in B-ISDN/ATM. In *Proc. Fifth Intl. Conf. Data Comm. Sys. and Their Perf.*, Raleigh, NC, October 1993.
- [6] K.Y. Eng, et al. BAHAMA: A Broadband Ad-Hoc Wireless ATM Local-Area Network. *Proc. ICC'95* (1995) pp. 1216-1223.

- [7] G.H. Forman, and J. Zahorjan. The Challenges of Mobile Computing. *IEEE Computer*. pp. 38-47. April 1994.
- [8] E. N. Gilbert. Capacity of a Noise-Burst Channel. *The Bell System Technical Journal*. pp. 1253-1265. September 1960.
- [9] J. Hagenauer, N. Seshadri, and C.W. Sundberg. "The performance of rate-compatible punctured convolutional codes for digital mobile radio." *IEEE Transactions on Communications*, COM-38(7): 966-980, 1990.
- [10] M.J. Karol, Z. Liu, and K.Y. Eng. An Efficient Demand-Assignment Multiple Access Protocol for Wireless Packet (ATM) Networks. *Wireless Networks* 1 (1995) pp. 267-279.
- [11] M. Naghshineh, M. Schwartz, and A.S. Acampora. Issues in Wireless Access Broadband Networks. IBM Research Report 19980, November 1994.

A Review of Call Admission Control Schemes in Wireless ATM Networks

Dimitrios D. Vergados, Nikolaos G. Protopsaltis, Christos Anagnostopoulos,
John Anagnostopoulos, Michael E. Theologou and Emmanuel N. Protonotarios

Telecommunications Laboratory
Department of Elec. Engineering & Computer Science
National Technical University of Athens
Herion Polytechniou 9, Zographou Athens
GR 157 73, Greece
Tel: +30 1 772 2558
Fax: +30 1 772 2534
{vergados; nproto}@telecom.ntua.gr

Abstract. The next generation personal communication networks are expected to support multimedia services and a general solution for this is viewed through wireless ATM. The requirements to provide real time multimedia traffic along with voice and data of varying priorities increase the need for QoS mechanisms in wATM. The problem of Admission Control is one of the most important issues in a Wireless ATM environment. This paper reviews significant strategies, as new algorithmic schemes for Call Admission Control (CAC) in Wireless ATM Networks and evaluates their results.

1 Introduction

Wireless ATM technology has the ability to deliver high degrees of quality of service (QoS) while supporting multiple traffic classes on the same transmission path [5], [7]. In this context, congestion control through adequate buffering is becoming particularly significant to minimize the probability of cell loss and cell delay when multiple large traffic bursts are received concurrently at a switch.

Wireless ATM is a connection-oriented service. One significantly important area in Wireless ATM networks is the control of the network congestion. Its principal role is to protect the network and the user in order to achieve network performance objectives and optimize the usage of network resources [9]. Congestion control procedures can be classified into preventive control and reactive control. Preventive congestion control involves the following two procedures: call admission control (CAC) and bandwidth enforcement. Before a user starts transmitting over a Wireless ATM network, a connection has to be established which is achieved at call setup. This Virtual Path between the sender and the receiver involves one or more ATM switches. On each of these ATM switches, resources have to be allocated to the new connection. The resource manager of the switch among other management functions accept/reject the new connections or tears down old ones. If the new connection is accepted during a new call or a handoff procedure we have a consequent bandwidth and/or buffer allocation, which are released when the connection is terminated or base

station is changed. The CAC deals with question of whether or not the switch can accept the new connection.

Admission control ultimately decides whether to admit or reject the request to add a new flow or connection based upon whether the newcomer would violate delivering QoS for existing flows or connections [10]. Furthermore, Admission Control involves each node checking every request against available capacity and current QoS capabilities. The node admits the request only if can provide the requested QoS after adding the traffic corresponding to the existing wireless connections. In our paper the QoS issues related to the Call Admission Control in wireless ATM Networks are discussed analytically. Therefore, our paper is organized as follows: In the following an extensive overview of the problem formulation regarding the QoS in Wireless ATM is taking place. In section 3, the presentation of these algorithms is taking place and the evaluation results are discussed. Finally, the last section has the conclusions for the aforementioned reviewed schemes.

2 Modeling Wireless ATM Network – Problem Formulation

Wireless ATM networks that support multimedia services with a QoS mechanism provide some challenges that are also met in cable networks, such as mobility, routing information accuracy, scalability, interoperability and adaptability [12], [13]. Although, delivering hard QoS guarantees in the wireless domain is rather difficult since assumptions made in providing QoS guarantees in wired ATM networks do not always hold in their wireless extension due to large-scale mobility requirements, limited radio channel resources and fluctuating network conditions.

Qualities of Service guarantees are one of the main advantages envisaged for ATM networks. While other networks such as IP networks do not guarantee QoS, ATM Networks do at the cost of higher complexity [32]. Therefore Wireless ATM networks should provide mobile QoS composed of three different parts: Wired QoS, Wireless QoS and handoff QoS. The wired QoS consists of the following basic parameters link delay, cell delay variation, bandwidth, cell error rate etc [24], [30]. In the wireless QoS the error rate is now typically some order of magnitude higher. Also, channel reservation and multiplexing mechanisms at the air interface strongly influence cell delay variation. In the Handoff QoS new parameters are introduced such as handoff blocking, cell loss during handoff, handoff speed. Also, some other major components that consist Wireless ATM are the core network wired infrastructure and the Wireless access link. The network infrastructure provides the necessary mobility support (location management and connection handoff) to the end terminals. The prefix in each Wireless ATM terminal address is supplied by the switch to which the terminal is connected. An integrated scheme proposed in radio ports acts as switches with single or multiple Wireless ATM interfaces. The home address remains the same regardless of the mobile terminal's location. When the terminal reaches another radio port another network prefix is allocated to the terminal. This leads to a scheme where a mobile terminal maintains a virtual connection to its home switch. When the terminal changes location a handoff procedure arises to reroute the location update virtual connection to the terminal's new foreign switch. All the above ensure a successful location management in a wireless ATM network. On the other hand, the

connection handoff is a procedure where a user's radio link is transferred between radio ports in the network without any interruption for the user connection. Handoff minimizes interference to the users when they move through neighboring cells and ensures the integrity of the radio connections [11]. Handoff assists users to freely move and communicate beyond the limits of a Wireless specific area. The connections of a user terminal in a Wireless ATM network may need to be rerouted in cases of handoff [22].

Therefore the Admission Control Mechanism during a call setup or a handoff procedure is very important for Wireless ATM Networks [8], [19]. Call Admission Control (CAC) is a function commonly implemented by software in wireless ATM switches that determines whether to admit or reject connection requests. A connection request includes traffic parameters, along with either the ATM Service category, requested QoS Class, or the user specified QoS parameters [27], [28]. Wireless ATM switches use CAC to determine whether admitting the connection request at Permanent Virtual Connection (PVC) provisioning time or Switched Virtual Connection (SVC) call origination time would violate the QoS already guaranteed to active connections. CAC admits the request only if the network can still guarantee QoS for all existing connections after accepting the request [7], [27], [31]. Frequently, each node performs CAC for SVCs and Soft Permanent Virtual Connections (SPVCs) in a distributed manner for performance reasons. A centralized system may perform CAC for PVCs. For accepted requests, CAC determines policing and shaping parameters, routing decisions, and resource allocation. CAC must be simple and rapid to achieve high SVC call establishment and on the other hand CAC must be accurate to achieve maximum utilization while still guaranteeing QoS [25]. CAC complexity is related to the traffic descriptor, the switch queuing architecture and the statistical traffic model. In general a network uses the peak cell rate, sustainable cell rate, and the maximum burst size for the two types of CLP flows, as defined in the traffic contract to allocate the buffer, trunk and the switch resources. Packet rate allocation ensures that even if all sources send the worst-case, conforming cell streams, the network still achieves the specified Quality of Service (QoS). Similar CAC Algorithms using the SCR and MBS parameters also achieve the lossless multiplexing [26]. CAC implementations may also permit a certain amount of resource oversubscription in order to achieve statistical multiplex gain. CAC algorithms may also use a concept called equivalent capacity in an admission algorithm based upon combination of the PCR, SCR, and MBS. The basic Traffic Management functions of a Wireless ATM network is shown in the following figure.

3 Call Admission Control Schemes and Algorithms Evaluation

As it is mentioned earlier the CAC decides on whether a new connection is going to be accepted or not. This is based on the influence to the QoS that the new connection is going to have and to if the switch can provide the required QoS to the new connection [9], [11]. The main CAC schemes are classified to the peak bandwidth allocation and statistical allocation. The peak bandwidth allocation which is based on the constant bit rate (CBR) services is suitable for PCM-encoded voice, other fixed-rate applications, unencoded video and other very low bandwidth applications (telemetry). The advantage of peak bandwidth or nonstatistical allocation is that the

decision of accepting or not a new connection is very easy to be taken [1], [2]. If the sum of all existing connections plus the rate of the new connection exceeds the capacity of the output link then the new connection is not accepted. The disadvantage is also obvious. If the connections do not transmit at peak rates the output port link is underutilized.

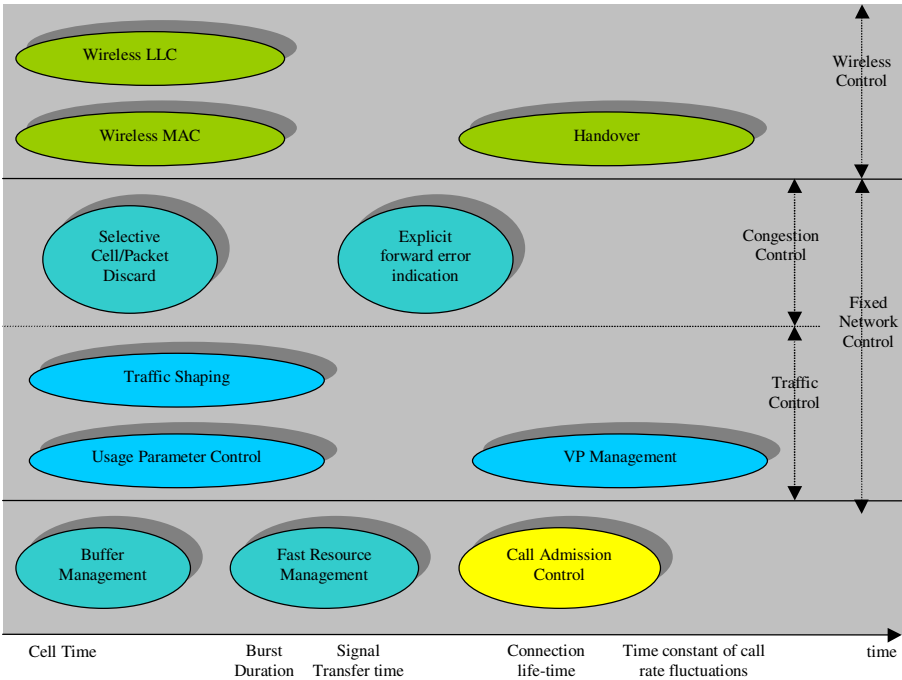


Fig. 1. Wireless ATM Traffic Management Functions

The problems that arise with Statistical allocation is that the characterization of an arrival process and how this is shaped deep in the ATM network is difficult to be done [4]. Furthermore, decisions must be made on the fly and thus, may not be CPU intensive. The arrival process has been characterized by using Poisson processes, its discrete counterpart Markov modulated Bernoulli and a fluid processes. Several auto-regressive models have been proposed to characterize traffic due to video. The theory of self – similarity includes long – term correlation in the arrival process. The appropriateness for models in ATM traffic is based on a few parameters standardized by the ATM Forum. These parameters are the peak rate, the average rate, the cell delay variation for the peak rate and the maximum burst length. These are not adequate when it comes to bandwidth allocation. Burstiness and correlation are the two parameters that affect the QoS.

The algorithmic schemes described by [33] for presentation purposes are going to be based on a non-blocking ATM switch where congestion takes place in the output ports. A variety of different call admission schemes have been proposed in the literature. Some of these schemes require an explicit traffic model and some only require traffic parameters such as the peak and average rate. In this paper we review some of these CAC schemes. The schemes have been classified into the following groups:

- Equivalent capacity
- Heavy traffic approximation
- Upper bounds of the cell loss probability (CLP upper bound)

The equivalent capacity scheme is based on a single source that feeds a finite capacity queue. Then, the equivalent capacity of the source is the service rate of the queue that corresponds to a predefined cell loss. The equivalent capacity methods are inaccurate in some situations. Similarly to the equivalent capacity method that is based on the asymptotic behavior of the tail of the queue length distribution there is a proposed approximation for bandwidth allocation based on the same asymptotic behavior. The upper bounds cell loss probability is based on the average number of cells that arrive during a fixed interval and the maximum number of cells that arrive in the same fixed interval.

This classification was based on the underlying principle that was used to develop the scheme. In comparing these algorithmic schemes to each other, we focus on system throughput and class independence. While testing the effect of the buffer size the cell loss probability algorithm (CLP upper bound) seems to be the less sensitive. When we change the required cell loss probability we find out that the least sensitive is the equivalent capacity scheme while the heavy traffic is the more sensitive. The CLP upper bound seems to be sensitive as well. The algorithms differ in handoff admission. The equivalent capacity scheme algorithm completely shares available bandwidth among all arriving handoff users while the other algorithm uses a dynamic measurement-based reservation scheme to ensure that handoff users of each class achieve the required QoS. Also the algorithm formulated applies to a two or three-dimensional network topology of arbitrary shape. It is assumed here that the network is uniform, i.e., the movement of the mobile is independent of location and direction. As such, the probability of handing off to or from any cell is the same as any other cell. The other two algorithms that were introduced in [33] differ from the previous in that they use dynamically adjusted reservation partitions to control the blocking probability profile. The third algorithm is termed independent multimedia one-step prediction, complete sharing variant, and the second fourth is independent multimedia one-step prediction, reservation variant. The mechanism used to control the relative call blocking probabilities is based on a performance measurement function and is updated periodically.

Another Admission Control Scheme is proposed by [15] using Genetic Algorithms. A Genetic Algorithm (GA) starts with an initial population that consists of possible solutions to the optimization problem the so – called individuals, and thereafter generates better one population to find the optimal solution [3].

The simple genetic Algorithm (SGA) consists of four components:

- Initialization
- Evaluation of the fitness function
- Selection
- Genetic Operators

The initialization is based on seeds that are selected randomly or excellent seeds. These seeds are from an alphabet of floating point numbers with values within variables upper and lower bounds. Every individual is evaluated to obtain its fitness function. For the selection purposes a normalized geometric selection is used. Two genetic operators, the mutation and crossover are used. Application of these operators

is for the individuals. Mutation introduces new information to the population while crossover doesn't. This characteristic of the mutation allows it to overcome local optima. On the other hand, crossover involves the exchange of portions of two selected individuals. The genetic Algorithm is dedicated to the allocation of fair bandwidth shares and after that takes advantage of any available bandwidth left from this allocation. In order to obtain an efficient solution the proposed features are based on generation of then initial population, fitness function for fair bandwidth allocation and for any other available bandwidth and a decomposition scheme if the number of calls is too large. The simulation results from a video component of a multimedia call for this algorithm show a significant gain in the minimization of the probability of having a call rejected and delivery of acceptable QoS to the users. Moreover the adoptive nature of the algorithm gives the advantage of increasing the QoS levels of the existing calls when one or more calls depart from the system.

Another algorithm regarding Call Admission Control (CAC) in a wireless ATM network is described in detail by [6]. This algorithm is based on a threshold-based mechanism in a wireless ATM environment, which decides the admission of calls using the Available Bit Rate (ABR) mode, standardized by the ATM Forum for transmission on wired ATM networks [14], [20]. The mechanism used a threshold to privilege handoff requests over new call requests. Similarly to the ABR, each user declares a Minimum Cell Rate (MCR) and a Peak Cell Rate (PCR) for each media. The admission controller, one for each cell in the wireless access network, is an explicit rate switch, implementing the third-generation ABR strategy, the Explicit Rate (ER), by which it is possible to indicate directly the bandwidth that can be assigned to the connection by means of the ER field in the Resource Management (RM) cells. That system is a wireless cell, belonging to a cell cluster, with a specified number of channels in each cell; the input traffic comprises both new calls and calls coming from neighboring cells in the same cluster due to handoff. According to this scenario scheme, users arrive in the cell from the neighboring cells with a Poisson distribution with different arrival rate of the requests due to handoff and arrival rate of the requests due to new calls. A call can be accepted only if the network can guarantee the minimum requirements. If, on the other hand, there are free channels, they are assigned, according to a proportional strategy, to the calls that before were transmitting with a number of channels less than the PCR. The system performance evaluated in two different cases with different number of channels in each cell was studied. It was shown that a small increase in the number of cell channels determines great differences in system performance. Also, it was shown that all the curves exhibit a staircase shape, due to the fact that small variations of the thresholds do not affect system behavior. As was to be expected, the new-call blocking probability decreases when the threshold increases, but to the expense of the handoff loss probability. The curves coincide when the new-call admission threshold is equal to the number of channels, that is, when no priority is given to active calls coming from neighboring cells due to handoff. Finally, it was observed that the larger the number of users, the worse the QoS will be in terms of both blocking and loss probabilities. Thus, the performance of that strategy was evaluated by analyzing the influence of the system parameters in a case study. It has been concluded that according to the number of users involved in a given cell, of the threshold to be applied to improve the call loss probability due to handoff, limiting the number of calls simultaneously active.

Another scheme for the Admission Control in Wireless ATM Networks is proposed by [23] in order to maintain the QoS traffic requirements for potential

handoff calls. According to this scheme, the system decisions are based on the developed jitter bounds for (CBR) traffic and delay bounds for (VBR). The wireless ATM Network consists of BSs supporting different types of traffic sending and receiving packets to and from all users through the downlink and uplink channels, the MSC and of course the ATM backbone network. In the three ATM traffic types and in the classes inside the types CBR, VBR, ABR different priorities are given. More specifically CBR and the ABR have the highest and lowest traffic transmission priority respectively. Also, inside the classes with smaller jitter and delay respectively tolerance take higher priority. In VBR each call is regulated by a leaky bucket (LB). Therefore packet transmission is directed by BS according to the preset priority. In the proposed CAC scheme before an admission decision is taking place during a handoff or a new call, the QoS performance bounds are checked. The simulation results shown that the proposed CAC scheme can achieve both low handoff call dropping rate and high resource utilization. For this CAC Scheme the deterministic bounds for the nonpreemptive polling process have been used for QoS provisioning. It is anticipated that the performance should be further improved with stochastic bounds.

4 Conclusion - Brief Discussion

Wireless communications have undergone a tremendous growth in recent years. With systems for mobile analog and digital cellular telephony, radio paging, and microwave/satellite broadcasting becoming widespread, next generation wireless communications systems such as wireless ATM (WATM) will be required to support the seamless delivery of voice, video and data with high quality. Delivering hard QoS guarantees in the wireless domain is rather difficult since assumptions made in providing QoS guarantees in wired ATM networks do not always hold in their wireless extension due to large-scale mobility requirements, limited radio channel resources and fluctuating network conditions.

In cellular-based wireless networks, the quality of service (QoS) provisioning problem is more challenging due to wire-less channel fading, bit error rate (BER), and mobility. Fading, in addition to BER, causes packet loss and delays due to retransmissions. Bandwidth availability is highly un-predictable due to time and spatial dependencies, in addition to fading effects. During handoff, a mobile that was granted certain QoS guarantees could be deprived of such guarantees or even dropped altogether. Hence, the bandwidth allocated to a call, during setup phase, could be decreased significantly during the call lifetime.

Therefore, a significant contribution of our approach to CAC is the consideration of two conflicting goals: to support the dynamic and transient nature of device mobility and QoS adaptation; and to limit the disturbance incurred by QoS renegotiation and handoff. This paper evaluates efficient algorithms for managing and controlling Wireless ATM networks, which are a key prerequisite for the successful deployment of this networking technology. Several research initiatives aim at enabling researchers and engineers to understand the traffic characteristics and their impact on control mechanisms through measurement, simulation and analytical studies. Further research work should be done and evaluated in the future as

Admission Control is still an open and important issue for the successful development of wireless communications and especially for the wireless ATM networks.

References

1. Bora Akyol, and Donal C. Cox, , "Signaling Alternatives in a Wireless ATM Network, "IEEE JSAC, Vol. 15, No. 1, pp. 35-49, Jan. 1997.
2. D. Raychaudhuri and D. Wilson, "ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks ", IEEE Journal On Selected Areas In Communications, Vol. 12, No 8, Oct. 1994, pp 1401 - 1413.
3. D. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning. Reading, MA: Addison Wesley, 1989.
4. L. Fernandes, " Developing a System Concept and Technologies for Mobile Broadband Communications", IEEE Personal Communications, Feb. 1995, pp 54 - 59. Trans. Networking, pp. 1-12, 1997.
5. Anthony Acampora, "Wireless ATM: A Perspective on Issues and Prospects," IEEE Personal Communications, pp. 8-17, Oct 1996.
6. A. Lombardo, S. Palazzo and G. Schembra, "An Admission Mechanism for ABR Traffic in an ATM Wireless Network", Nec Research Institute, 1999.
7. D. McMillan, "Traffic Modeling and Analysis for Cellular Mobile Networks", *Teletraffic and Datatrafic, Elsevier Science Publishers (North Holland)*, IAC, 1991.
8. H. Saito and K. Shiimoto, "Dynamic Call Admission Control in ATM Networks", IEEE J. on Selected Areas in Communications, Vol. 9, Sept. 1991, pp. 982-989.
9. C. Chao and W. Chen, "Connection admission control for mobile multiple class personal communications networks," IEEE JSAC, vol. 15, pp. 1618-1626, Oct. 1997.
10. M. Bracha Epstein and Mischa Schwartz, Predictive QoS-Based Admission Control for Multiclass Traffic in Cellular Wireless Networks IEEE JSAC, Vol. 18, No. 3, Mar. 2000 523.
11. A. Acharya, J. Li, and D. Raychaudhuri, "Primitives for Location Management and Handoff in Mobile ATM Networks", ATM Forum Contribution 96-1121, August 1996
12. B. Epstein and M. Schwartz, "QoS-based predictive admission control for multi-media traffic," in Broadband Wireless Communications, M. Luise and S. Pupolin, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 213-224.
13. R. Ramjee, D. Towsley, R. Nagarajan, "On Optimal Call Admission Control in Cellular Networks", *Baltzer, Wireless Networks*, Vol. 3, 1997.
14. H. Saito, K. Kawashima, H. Kitazume, A. Koike, M. Ishizuka, and A. Abe, "Performance Issues in Public ABR Service", *IEEE Communications Magazine*, Nov. 1996.
15. M. R. Sherif, et al., "Adaptive Allocation of Resources and Call Admission Control for Wireless ATM Using Genetic Algorithms", IEEE JSAC, Vol. 18, No. 2, Feb. 2000.
16. N. Naghshineh, M. Schwartz, "Distributed Call Admission Control in Mobile/Wireless Networks", *IEEE Journal of Select. Areas Comm.*, Vol. 14, No. 4, May 1996.
17. Ender Ayanoglu, Kai Y. Eng and Mark J. Karol, "Wireless ATM: Limits, Challenges, and Proposals," IEEE Personal Communications, pp. 18-34, Oct 1996.
18. J.Porter and D. Gilmurray, "Tunnelled Signaling for the Support of Mobile ATM", ATM Forum Technical Committee, ATM Forum/96-1699, Dec., 1996.
19. ATM Forum, ATM Traffic Management Specification, Version 4.0, February 1996.
20. R. Jain, S. Kalyanaraman, S. Famhy and R. Goyal, "Source Behavior for ATM ABR Traffic Management: An Explanation", *IEEE Communications Magazine*, Nov. 1996.
21. B. Jabbari, G. Mason, "Teletraffic Aspects of Evolving and Next-Generation Wireless Communication Networks", *IEEE Personal Communications*, Dec. 1996.

22. B. A. Akyol, D. Cox, "Rerouting for Handoff in a Wireless ATM Network", *IEEE Personal Communications*, October, 1996.
23. D. Zhao, X. Shen and J.W. Mark, "Efficient Call Admission Control for Heterogeneous Services in Wireless ATM Networks, *IEEE Commun. Magazine*, Oct. 2000.
24. S. Jordan, P. Varaiya, "Control of Multiple Service, Multiple Resource Communication Networks", *IEEE Transactions on Communications*, Vol. 42, No. 11, Nov. 1994.
25. R. Bolla, F. Davoli, M. Marchese, "Simple Schemes for Traffic Integration at Call Set-up Level in ATM Networks", *Computer Communications*, Vol. 19, 1996.
26. E. Gelenbe, X. Mang and R. Onvural, "Bandwidth Allocation and Call Admission Control in High Speed Networks", *IEEE Comm. Magazine*, May 1997.
27. M. Guizani and A. Rayes, "Designing ATM Switching Networks", McGraw-Hill, 1998.
28. D. McDysan and D. Spohn, "ATM Theory and Applications", McGraw-Hill, 1998.
29. J. Schiller, "Mobile Communications", Addison-Wesley, 2000.
30. C. Perkins, "Ad Hoc Networking", Addison-Wesley, 2001.
31. B.H. Walke, "Mobile Radio Networks", Wiley, 1999.
32. D. McDysan, "QoS and Traffic Management in IP & ATM Networks", McGraw-Hill, 2000.
33. Harry G. Perros, "Call Admission Control Schemes: A Review", *IEEE Comm. Magazine*, Nov. 1996

Analysis and Evaluation of QoS-Sensitive Multicast Routing Policies*

Elena Pagani and Gian Paolo Rossi

Dip. di Scienze dell'Informazione, Università degli Studi di Milano
via Comelico 39, I-20135 Milano, Italy
{pagani, rossi}@dsi.unimi.it

Abstract. The provision of *Quality-of-Service* to multimedia applications in Internet is currently a hot research topic. Many approaches, architectures, models and protocols have been proposed in the literature. The proposals impact on the whole system architecture. So far, only a few of them have been implemented and tested. The standardization process is far to be completed. In this paper, we discuss and compare three multicast routing algorithms that have been proposed in the literature, aiming at characterizing tree structures according to the QoS constraints imposed by either the traffic sources or the recipients. Two of those algorithms have been evaluated by means of simulations.

Keywords: Quality-of-Service, admission control, multicast routing, performance evaluation.

1 Introduction

The deployment of protocols and architectures supporting *Quality-of-Service* (QoS) in IP networks is strongly demanded by the growing diffusion of multimedia and real-time applications for the Internet. Those protocols must provide multicast support to fulfil the requirements of a large part of the applications, characterized by a 1-to-many or many-to-many interaction schema. Many proposals have been presented in the literature, aiming at either defining QoS-sensitive functional architectures [1,2,3], or providing possible implementations for the architectural modules [4,5]. So far, however, few proposals have been realized and tested.

In this work, we analyze three approaches proposed in the literature to perform the admission control for QoS multicast traffic. Those approaches allow to characterize tree structures for the forwarding of QoS traffic, in either int-serv domains or access networks to diff-serv backbones. We compare the performance of these approaches by means of simulation techniques. The paper is structured as follows: in Section 2, we introduce the system model considered throughout the work. In Section 3, we describe the different approaches proposed in the

* This work was supported by the MURST under Contract no.MM09265173 "Techniques for end-to-end Quality-of-Service control in multi-domain IP networks".

literature to the QoS-sensitive multicast routing problem, focusing in particular on the QoSMIC [6], QoSCBT [7] and QoSPIM [8] algorithms. In Section 4, we analyze the pros and cons of those three protocols, while in Section 5, we discuss the simulation results obtained by implementing QoSMIC and QoSCBT in the framework of the NS-2 simulation package [9]. Some concluding remarks are presented in Section 6.

2 The System Model

A network $\mathcal{N}$ can be represented as a weighted graph $G = \langle V, E \rangle$ where the vertices in V correspond to the network routers and the edges in E correspond to the links connecting the routers. Let $N = |V|$ be the number of routers in $\mathcal{N}$. The weight of an edge e is a vector of n elements. Each element represents the quality of the link in terms of a different metric. Examples of metrics are the bandwidth, the transmission delay and the packet loss probability. The metrics may be either *static* or *dynamic*. The static metrics (e.g., link capacity, reliability) only depend on the network technology. By contrast, the dynamic metrics (e.g., bandwidth availability, current queueing delay) depend on the current network status. The static metrics do not reflect the current traffic situation; considering them for the QoS routing may lead to inaccurate decisions, thus failing in providing the requested service level. On the other hand, the maintenance of updated estimates of the dynamic metrics can be costly.

The weight of a path p between two nodes $u, v \in V$ is computed from the weights of the links forming p . Two weight elements compose differently, depending on whether the corresponding metric is *additive* (such as the delay), *multiplicative* (such as the loss probability) or *concave* (such as the bandwidth) [7]. In this work, we indicate with $m(u, v)$ the value of the metric m along the path from u to v provided by the unicast routing protocol.

An application s generating QoS traffic uses a session announcement protocol (e.g., **sdr** [10]) to announce the needed session information such as the transmission start time and the multicast address of the destination group. According to the int-serv model [2], a host h that wants to receive the s ' flow joins the corresponding destination group $\mathcal{G}_s$: h requests to graft the distribution tree $\mathcal{T}_s$ for $\mathcal{G}_s$ and it specifies the QoS it wants to receive, represented as a vector req_h of n elements. In this paper, we only focus on the int-serv model, because in the diff-serv model [1] many problems arise concerning the support to dynamic changes of the multicast group membership [11], that deserve further study.

3 Proposed QoS Multicast Routing Policies

Several algorithms have been recently proposed in the literature, to compute tree structures to forward QoS multicast traffic. They differentiate in the number and nature of the considered metrics, and in the methods used to characterize the tree. The (sub)optimal tree may be searched using *heuristics*, according to

the result of NP-completeness for the problem of computing an optimal path considering more than one additive metric constraint [8].

Some proposals [12,13,14] present algorithms that exploit the knowledge of the network topology and resource availability, the sources, destinations and resource requirements of the QoS traffic, to search for a tree satisfying the QoS constraints. Part of the needed information may be obtained for instance from an underlying link-state protocol. However, the specifications of those algorithms usually disregard the implementation issues. Neither the system architecture nor the communication pattern among the involved parties are detailed.

In this paper, we focus on algorithms which have a distributed control, possibly derived by adding QoS-awareness to standard multicast routing protocols. In particular, we have studied the behaviours of the QoSMIC [6], QoSCBT [7] and QoSPIM [8] multicast routing protocols. All these protocols aim at characterizing a tree structure $\mathcal{T}$ that connects the hosts in a group $\mathcal{G}$, and is *shared* among all the sources sending traffic to $\mathcal{G}$. Let $\mathcal{R}$ be the root of $\mathcal{T}$. $\mathcal{T}$ is such that $\forall g \in \mathcal{G}$ a path p exists in $\mathcal{T}$ from $\mathcal{R}$ to g so that, for each metric m_i , $1 \leq i \leq n$, p satisfies $m_i(\mathcal{R}, g) \leq req_g[m_i]$ if m_i is either an additive or a multiplicative metric, or $m_i(\mathcal{R}, g) \geq req_g[m_i]$ if m_i is a concave metric. The algorithms can support *heterogeneous* recipients, that is, recipients having different QoS requirements. We describe them in detail in the next sections.

3.1 QoSMIC

According to the QoSMIC protocol [6], a new router nr joining $\mathcal{G}$ sends its request to $\mathcal{R}$, which forwards it along $\mathcal{T}$ to identify a subset of the nodes in $\mathcal{T}$ as the *nr candidates* to graft to the tree. Several heuristics can be used to characterize the set of candidates; we adopted the *multicast tree search* policy with the *directivity* mechanism for the selection of the candidates. A node $n_c \in \mathcal{T}$ is a candidate if the quality of the unicast path connecting n_c to nr , according to the static metrics, is such that the QoS requested by nr can be provided and n_c realizes a local optimum in terms of *quality-of-path*, that is, it offers a path whose quality is equal to or better than that offered by its ancestors in the tree. Each node in $\mathcal{T}$ carries its estimated QoS towards nr in the request forwarded along the tree, if it is better than that currently reported, to inform its descendants about it. The candidates send a **bid** message to nr ; the **bid** is used to evaluate the quality-of-path according to the dynamic metrics, and establishes a tentative reservation in the traversed nodes. nr waits for an interval Δt to receive the **bids** from all the candidates. Basing on the quality estimates carried by the received **bids**, nr chooses the router (and the branch) that offers the best QoS. nr sends to the chosen candidate a **graft** message, that establishes the resource reservation in the traversed nodes. The tentative reservations for the other candidates are removed upon a timer expiration. **graft** messages are periodically re-sent to refresh the reservations. QoSMIC can consider only one *quality-of-path* metric at a time (i.e., $n = 1$); it requires an underlying QoS-sensitive unicast routing protocol that finds “good” paths according to that metric. As a router only uses

the static metrics to decide whether it is a candidate, no status information exchange amongst nodes is needed upon join/leave events.

3.2 QoSCBT

QoSCBT [7] can contemporarily consider multiple metrics. A host h wishing to join the group $\mathcal{G}$ sends its request on the unicast path toward $\mathcal{R}$, with its QoS requirements and the indication about whether it is only a recipient in $\mathcal{G}$ or it may be also a source for $\mathcal{G}$. Depending on this information, the join request is processed differently. Each intree node maintains status information concerning the dynamic metrics measuring the local resource availability. The status is updated according to a *soft-state* approach via the periodic exchange amongst neighbours of control messages, that also refresh the resource reservations. An intree router r records:

- for each additive metric m_a : both the maximum value of m_a from r to any downstream destination, $M_D^r[m_a] = \max\{m_a(r, dest) \forall r's \text{ downstream destinations}\}$, and the maximum value of m_a from any downstream source to r , $M_S^r[m_a] = \max\{m_a(src, r) \forall r's \text{ downstream sources}\}$;
- similar information is maintained for multiplicative metrics;
- for a concave metric m_c : the local value of m_c for each downstream interface, and the number of downstream sources.

A join request is caught by the first intree node r it encounters while traveling towards $\mathcal{R}$, and it is processed as follows:

- for an additive metric m_a : the request is accepted if $M_S^r[m_a] + m_a(r, h) \leq req_h[m_a]$ and, if h is also a source for $\mathcal{G}$, $M_D^r[m_a] + m_a(h, r) \leq req_h[m_a]$. Similarly for multiplicative metrics;
- for a concave metric m_c : the request is accepted if $m_c \geq req_h[m_c]$ for the r 's downstream interface to h .

If the request is rejected, a notification is sent to h . Otherwise, for additive and multiplicative metrics the request is forwarded up to $\mathcal{R}$, with each intermediate router repeating the test. In case of success, $\mathcal{R}$ sends an acknowledgement to h ; each router that receives the ack performs the appropriate resource reservation, and updates its state if needed. For concave metrics, if h is not a source the request is immediately acknowledged by the first router r performing the test, which allocates the requested resource. Otherwise, if r has not any other downstream source, it installs a tentative state and forwards the request upstream to perform the resource reservation. The procedure is repeated until an ancestor is encountered, which already has downstream sources, or the request is rejected. The described mechanism works for many-to-many applications with one active source at a time; this way, the reservation can be shared among all the sources. To avoid conflicts, QoSCBT imposes that a node processes only one join request at a time; other requests received in the meantime stay waiting until the (n)ack for the previous request has been sent.

3.3 QoS PIM

In [8] two techniques are proposed to extend the PIM-SM protocol [15] adding QoS-awareness. In this work we focus in particular on the Tree Information-Based QoS Multicast (TIQM) protocol. TIQM may contemporarily satisfy bandwidth, delay and loss constraints. It exploits the service of an underlying *link state* protocol, that maintains updated information about the network topology and the resource availability. A host h that wants to join a group $\mathcal{G}$, uses the link-state information to compute a subgraph $G' \subseteq G$ obtained from G by pruning the links that do not have enough available bandwidth. Then, h runs the Dijkstra's algorithm to characterize all the paths in G' connecting it to the nodes in $\mathcal{T}$, and having a delay lower than $req_h[delay]$. h chooses the minimum delay path. As the link state protocol, although maintaining the dynamic metrics, could be temporarily inaccurate, h tests the current delay and loss metrics for that path, by sending a join request along the path using source routing. When the request arrives at an intree node, it is forwarded upstream along the tree up to $\mathcal{R}$, and a local reservation is recorded if needed. Every router traversed by the request checks for the actual availability of the requested resources and performs the reservation. If the check fails, a negative ack is sent to h along the reverse path, to release the reserved resources. Otherwise, when the request is received by $\mathcal{R}$, $\mathcal{R}$ sends an ack to h .

4 Analysis of the Described Algorithms

In Table 1 we summarize the main characteristics of the studied protocols. The main drawback of QoS MIC is its dependence on a QoS-sensitive unicast routing. Both the QoS MIC and the QoS CBT protocols operate by checking whether enough resources are available along the path provided by the unicast routing between an intree router and the joining node. Since the unicast routing considers only static metrics¹, the consequence of that policy is that flows tend to converge on the paths with greater resource availability. When those paths are congested, no new service can be established until ongoing transmissions terminate and release the used resources. To some extent, QoS MIC overcomes this problem, thanks to its capability of testing alternative branches starting from different candidates.

A critical aspect of QoS CBT is how the router status is initialized. For some metric (e.g., the bandwidth availability), the initial status equals the static metric. The status is updated by each accepted join request, that results in a local resource reservation. Hence, dynamic metrics can be locally maintained. Other metrics, such as the delay, dynamically vary according to the traffic conditions. In [7], the authors propose to initialize the router status using the information gathered by the join request along its path. If that information is the quality-of-path experimented by the request itself, it concerns the direction *opposite* to that followed by the QoS data, and it does not provide any information about

¹ Dynamic metrics are evaluated during the tree construction procedure.

the traffic congestion on the path towards the joining node. If, by contrast, every router has to measure and piggyback over the request the estimates of its own quality-of-path towards the joining node, this charges the routers with an unacceptable overhead. Moreover, not all the routers could be capable of performing those measurements. On the other hand, the inaccuracy in estimating the queueing delay causes the failure in providing the requested QoS.

In QoSCBT, the lack of concurrency in managing join requests may increase the set-up latency observed by the recipients, particularly at the start of a session when the most part of the receivers join at almost the same time. Another QoSCBT drawback concerns its suitability limited to applications having one active source at a time.

Table 1. Comparison among QoSMIC, QoSCBT and QoSPIM

	QoSMIC	QoSCBT	QoSPIM
#metrics	1	≥ 1	≥ 1
unicast	same metric	independent	link-state
type of metrics	static/dynamic	dynamic	dynamic
multipath	yes	no	yes
tree	shared unidirectional source-based	shared bidirectional	shared unidirectional source-based
tentative state	soft	hard	hard/soft
sources	concurrent	one at a time	concurrent
concurrent JoinRq	yes	no	yes
reservation state	soft	soft	hard/soft
comm.complexity	$O(N)$	$O(N)$	$O(N^2)$

Both the QoSMIC and the QoSCBT communication overheads are $O(N)$. But, QoSCBT has a greater memory overhead because of the recording of the needed status information. The TIQM communication cost is $O(N^2)$ due to the use of a link state protocol. Moreover, it has the greatest computation and memory overhead. The update of the status information must be performed upon either each accepted join request or each leave event. On the other hand, the link state information provides the routers with many alternative paths that can be tested until one with the appropriate characteristics is found. In fact, we expect that TIQM outperforms QOSMIC in the probability of successfully grafting requesting recipients.

Both QoSMIC and TIQM may build either shared unidirectional trees or source-based trees. In the former case, additional mechanisms are needed to reserve resources along the unicast path from each source to the tree root.

5 Performance Evaluation

The above considerations are confirmed by the experimental results. We implemented the algorithms in the frame of the NS-2 simulation package [9]. We

simulated a meshed network of 64 nodes, connected by optical links of 2 Mbps bandwidth and variable length in the range 50 to 100 Km. QoS CBR traffic originates in the tree root; we performed our measures with different transmission rates. The background, best effort traffic is uniformly distributed all over the network and uses on average the 33% of the bandwidth resources. The packet size is 512B. Bandwidth reservation is ensured by using the WFQ [16] packet scheduling policy; the queue length is 20 packets. The DV-based unicast routing uses the link capacity as the quality-of-path metric. In the following, we report the results concerning QoSMIC and QoSCBT; the TIQM measurements are currently ongoing. We simulated an interval of 60 sec.

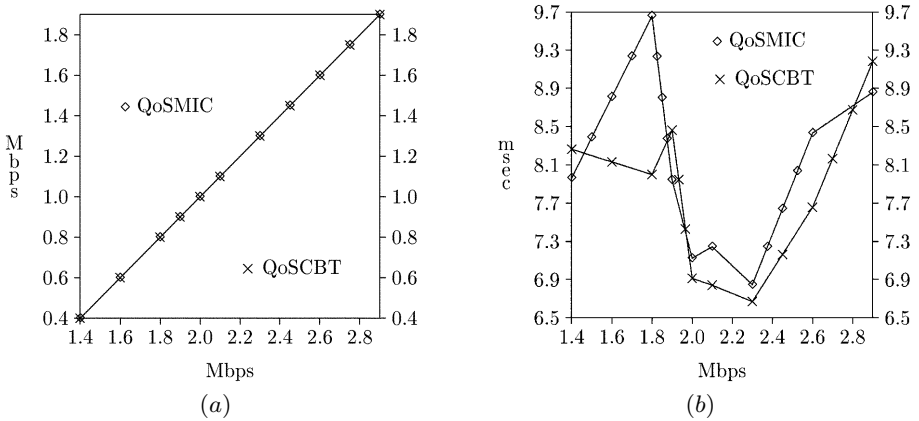


Fig. 1. (a) Throughput and (b) end-to-end delay of the second group vs. offered load, for $|\mathcal{G}| = 10$

We performed the first set of experiments with two QoS sources: the rate of the first source is 1 Mbps, while the rate of the second source varies between 0.4 and 1.9 Mbps. Both algorithms consider the bandwidth availability as the unique QoS constraint. Each source forms its own group; groups partially overlap. The second group is created after the first one has been established. We performed experiments with different membership cardinalities; however, the number of recipients has no impact on the delivered QoS.

Both algorithms guarantee the required bandwidth to the grafted recipients (figure 1(a)). For both algorithms, the end-to-end delay (figure 1(b)) increases when approaching the network congestion; while it drops when there is no space left for the best effort traffic (both source rates of 1 Mbps), thanks to the pipeline effect. When the second source rate increases, both algorithms successfully perform access control: only the destinations that can be grafted via non overlapped branches join the groups. Hence, on those branches, QoS traffic competes with the best effort traffic only. The observed fair delay and jitter are as well comparable for the two algorithms.

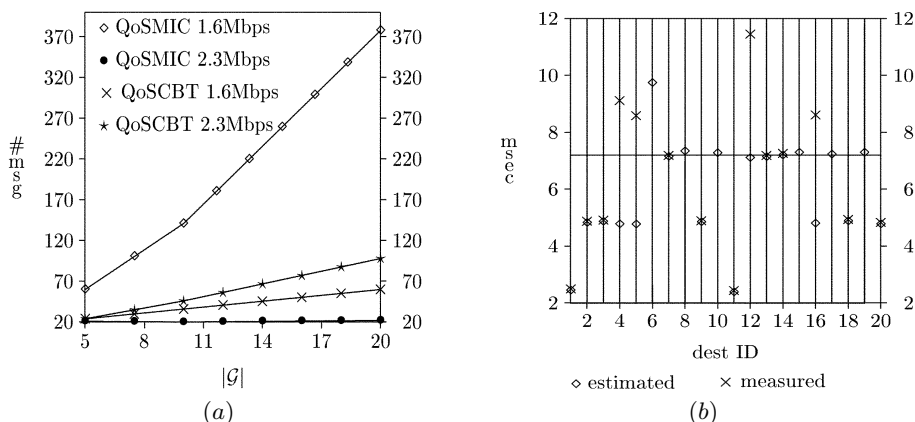


Fig. 2. (a) Number of control messages vs. number of destinations, for different QoS traffic loads. (b) Estimated and measured end-to-end delay for the different destinations, with delay constraint of 7.2 msec

By contrast, the algorithms differentiate in terms of costs. In QoS MIC, the need to wait for the bids sent by the candidates affects the latency observed by the joining nodes, if Δt is far greater than the round-trip delay between the joining node and the farthest intree node. In the opposite case, the joining node could not receive all the bid messages, thus possibly reducing its probability of successfully grafting the tree. QoS MIC also has a greater communication overhead than QoS CBT. In figure 2(a), we report the control messages generated to form the second source tree for both algorithms, with respect to the number of recipients. For low rate of the second source (0.6 Mbps), all the destinations are grafted. As QoS MIC characterizes several candidates, each of which sends its own bid, its overhead grows more than linearly with the number of recipients. On the other hand, for increasing rate, the testing of alternative branches allows QoS MIC to graft around 10% more recipients than QoS CBT. In the case of network congestion (offered load: 2.3 Mbps), very few intree nodes are candidates, and the QoS MIC overhead drops, differently from QoS CBT that has to send explicit rejection messages to remove the tentative reservations.

We performed experiments with QoS CBT considering both bandwidth and delay constraints. A request is accepted if the estimated transmission and propagation delay over the path from $\mathcal{R}$ to the joining node is lower than the required threshold. Queueing delays are not considered. In figure 2(b), we show the values of the estimated and measured end-to-end delay for each recipient, with only one source generating traffic at 1.6 Mbps, and uniform background traffic as before. With the considered threshold of 7.2 msec., 6 out of 20 receivers have not been grafted. Because of the contention with the best effort traffic and the consequent queueing delay, some destinations (4, 5, 12, 16) are accepted although their actual delay is greater than the threshold. The use of priority packet scheduling does not help in smoothing those peak queueing delays. If the delay

bound imposed by the application is strict, the implemented architecture must be enhanced by adding mechanisms to monitor the queueing delay. Anyway, the achieved delay profile is more homogeneous than without any delay constraint (figure 3(a)). This reflects as well on the fair delay: the lower the delay bound

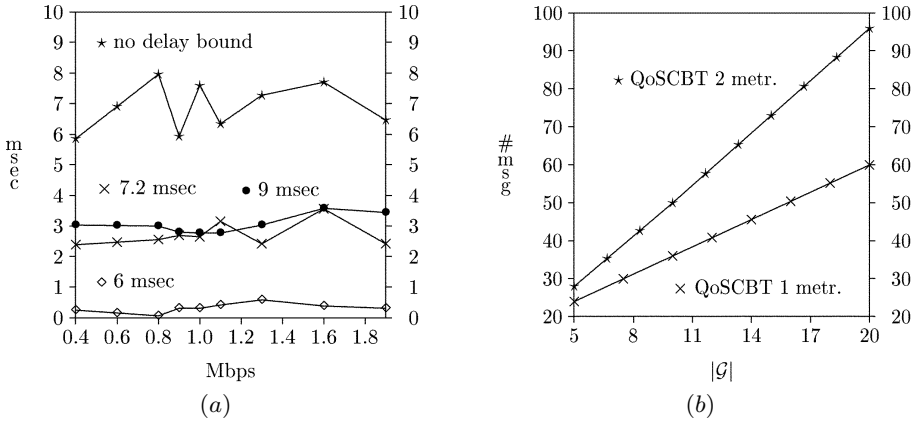


Fig. 3. (a) Average delay vs. source rate for $|\mathcal{G}| = 20$ and different delay constraints. (b) QoSCTB communication overhead vs. number of considered metrics, with source rate 0.6 Mbps and different group cardinalities

is, the lower the number of grafted recipients is, and the lower the difference among the delays they perceive is. On the other hand, as delay is an additive metric, the join requests are checked by all the nodes on the path from $\mathcal{R}$ to the joining node², thus increasing the communication overhead (figure 3(b)) and the latency (of roughly 1 msec.).

6 Concluding Remarks

In this work, we analyze the behaviours of three QoS-sensitive multicast routing protocols. All the algorithms build a shared tree, and are receiver-driven, thus supporting heterogeneous recipients. The study of their characteristics and the simulation results show that a trade-off exists between the algorithm costs and the probability of characterizing a suitable tree. In particular, this probability grows with the capability of exploring multiple paths towards a given receiver. Possibly, source-based trees could further increase that probability, by allowing a better traffic distribution all over the network, at the expenses of a larger status maintained by the nodes. We will investigate this issue by performing experiments with TIQM, that supports source-based trees as well.

All the algorithms require the modification of the current Internet routers. The choice of which algorithm to adopt depends on several issues, such as (i)

² Rather than only by the first encounteredintree node, as in the case of bandwidth.

the application characteristics (e.g., one or more sources contemporarily active) and requirements (e.g., one or more QoS constraints), and (ii) the dependences on the lower layer services.

Our future work involves the investigation of techniques to perform queueing delay estimation, so that QoS-CBT is more accurate in fulfilling the application requirements, without greatly increasing the overhead. Moreover, we are trying to reduce the dependency of QoSMIC and QoS-CBT from the unicast routing.

References

1. Blake S., Black D., Carlson M., Davies E., Wang Z., Weiss W.: An Architecture for Differentiated Services. RFC 2475, Dec. 1998. Work in progress.
2. Braden R., Clark D., Shenker S.: Integrated Services in the Internet Architecture: an Overview. RFC 1633, Jun. 1994. Work in progress.
3. Pagani E., Rossi G.P.: A Functional Architecture for End-to-end Quality-of-Service in a Multi-domain Network. Proc. 1st IEEE European Conference on Universal Multiservice Networks (ECUMN 2000), pp. 20-34, Oct. 2000.
4. Zhang L., Braden R., Berson S., Herzog S., Jamin S.: Resource reSerVation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205, Sep. 1997. Work in progress.
5. Schulzrinne H., Casner S., Frederick R., Jacobson V.: RTP: A Transport Protocol for Real-Time Applications. RFC 1889, Jan. 1996. Work in progress.
6. Faloutsos M., Banerjee A., Pankaj R.: QoSMIC: Quality of Service sensitive Multicast Internet protocol. Proc. SIGCOMM 1998, Sep. 1998.
7. Hou J., Tyan H.-Y., Wang B., Chen Y.-M.: QoS Extension to CBT. 9th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 99), Jun. 1999. (Also appeared as an Internet Draft, draft-hou-cbt-qos-00, Feb. 1999. Work in progress.)
8. Biswas S., Izmailov R., Rajagopalan B.: A QoS-Aware Routing Framework for PIM-SM Based IP-Multicast. Proc. GLOBECOM 2000, Nov. 2000. (Also appeared as an Internet Draft, draft-biswas-pim-sm-qos-00, Jun. 1999. Work in progress.)
9. Fall K., Varadhan K.: ns Notes and Documentation. The VINT Project, Jul. 1999. <http://www-mash.CS.Berkeley.EDU/ns/>
10. Handley M.: The sdr Session Directory: An Mbone Conference Scheduling and Booking System. <http://mice.ed.ac.uk/mice/archive/sdr.html>. Apr. 1996.
11. Bless R., Wehrle K.: IP Multicast in Differentiated Services Networks. Internet Draft, draft-bless-diffserv-multicast-01.txt, Nov. 2000. Work in progress.
12. Matta I., Guo L.: QDMR: An Efficient QoS Dependent Multicast Routing Algorithm. Journal of Communications and Networks, Vol. 2, No. 2, Jun. 2000, pp. 168-176.
13. Rouskas G.N., Baldine I.: Multicast Routing with End-to-end Delay and Delay Variation Constraints. IEEE Journal on Selected Areas in Communications, Vol. 12, No. 3, Apr. 1997, pp. 346-356.
14. Borella A., Cancellieri G.: Improving the Cost/Performance Ratio in Multicast Communications. Proc. NOC 2000, Jun. 2000, pp. 187-190.
15. Estrin D. et al.: Protocol Independent Multicast - Sparse Mode (PIM-SM). RFC 2362, Jun. 1998. Work in progress.
16. Shreedhar M., Varghese G.: Efficient Fair Queuing Using Deficit Round Robin. IEEE/ACM Transactions on Networking, Vol. 4, No. 3, pp. 375-385. Jun. 1996.

Multicast Performance of Multistage Interconnection Networks with Shared Buffering

Dietmar Tutsch¹, Matthias Hendler², and Günter Hommel²

¹ International Computer Science Institute

1947 Center St., Suite 600

Berkeley, CA 94704-1198, USA

DietmarT@cs.tu-berlin.de

² Technische Universität Berlin

Sekr. FR 2-2

Franklinstr. 28/29

D-10587 Berlin, Germany

{guppy,hommel}@cs.tu-berlin.de

Abstract. Multistage interconnection networks are often proposed to establish a multiprocessor system, ATM switches, or Ethernet switches. Various MIN structures exist to improve the performance. This paper investigates the buffer structures of shared and non-shared buffers in case of packet multicast. Shared buffers perform a dynamic buffer allocation but require a more complex switch management. The different behavior concerning uniform and non-uniform network traffic is examined. The simulation model copes with networks of arbitrary size, arbitrary switching element sizes, arbitrary buffer lengths in each network stage, and an arbitrarily chosen network load. Additionally, arbitrary multicast traffic patterns can be handled.

1 Introduction

Multistage interconnection networks (MIN) with the Banyan property are proposed to connect a large number of processors to establish a multiprocessor system [1]. They are also used as interconnection networks in Ethernet [18] and ATM switches [3]. Such systems require high performance of the network. To increase the performance of a MIN, Dias and Jump [4] inserted a buffer at each input of the switching elements (SE) and developed an analytical model to predict its performance. Buffers at each SE allow to store the packets of a message until they can be forwarded to the next stage in the network. In their model, Dias and Jump reduced each stage in the network to one SE of this stage so that it could be mapped to a Markov chain.

¹ Dietmar Tutsch is supported by the German Academic Exchange Service (DAAD) within the ICSI Postdoc Program

Jenq [8] introduced a model with lower complexity than that of Dias and Jump by considering only one input port of a SE per stage to model the complete stage. Yoon, Lee and Liu [17] extended Jenq's model by using arbitrary buffer lengths in the network and arbitrary SE sizes. Atiquzzaman and Akhtar [2] and Zhou and Atiquzzaman [19] examined nonuniform traffic like hot spot traffic. There are few investigations on multicast routing in MINs [9,10,12] and on the structure of multicast ATM switches [6,11]. An analysis of multicasting in MINs is presented by Yang [16]. But in contrast to the other models, this model is not able to deal with the backpressure mechanism to handle full buffers.

Tutsch and Hommel [14,13] extended Jenq's model such that the analytical model additionally copes with performance analysis of a network with multicasting. Multicasting includes the two special cases of unicasting and broadcasting of messages. Furthermore, the performance of MINs consisting of switching elements larger than 2×2 can be evaluated. In case of store and forward routing, a transient performance evaluation is available [15].

In this paper, a simulation model is used to investigate a shared memory approach in contrast to the previously mentioned network architectures. Packet multicast is taken into account. The paper is organized as follows. In Section 2, the architecture of a multistage interconnection network with shared buffers is introduced. The simulation model of such a network is developed in Section 3. This model is used to investigate the performance of shared and non-shared network buffers. Section 4 summarizes the research.

2 MIN Architecture

Simulation models of MINs allow a QoS (quality of service) comparison of various network architectures. MINs of special interest are such ones that connect multiprocessor systems or establish ATM and Ethernet switches. These internally clocked $N \times N$ MINs consist of $c \times c$ switches with $n = \log_c N$ stages (Figure 1). Internal clocking results in synchronously operating switches. In each stage k ($0 \leq k \leq n - 1$) of non-shared buffer networks, there is a FIFO buffer of size $m_{max}(k)$ in front of each switch input. The packets are routed by store and forward routing or cut-through switching from one stage to the succeeding by backpressure mechanism. Multicasting is performed by copying the packets within the $c \times c$ switches while routing (cell replication while routing, CRWR). Each packet copy is sent to the desired switch output independently of the other copies, even if another copy is blocked. These blocked copies are sent in the following clock cycles.

Networks consisting of shared buffers are established by replacing the c FIFO input buffers of size $m_{max}(k)$ of a $c \times c$ switch by one common buffer of size $c \cdot m_{max}(k)$ (Fig. 2). This shared buffer is organized as follows: Each switch input owns at least buffer space to store one packet avoiding the isolation of inputs (see below). The remaining buffer space of $c \cdot m_{max}(k) - c$ packets is available to all inputs. Each input forms a FIFO input queue of packets. If an input receives a new packet from the previous stage that has to be stored, the

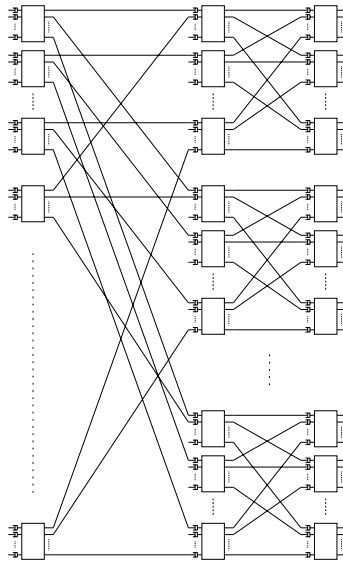


Fig. 1. 3-stage non-shared buffer MIN consisting of $c \times c$ SEs



Fig. 2. 4x4 switch consisting of a shared buffer (left) and non-shared buffer (right)

input allocates buffer space of the commonly used buffer part if available. If there is no further buffer available the packet is blocked at the previous stage.

An input with a queue of more than one packet deallocates buffer space if it sends a packet to the next network stage. This space is returned to the pool of the commonly available buffer.

Guaranteeing at least one buffer space to each input avoids that an input without any buffer cannot participate in the switch routing process because it is not able to receive a packet that has to be forwarded. E.g. let us assume that one of the inputs (hot spot input) receives much more packets than the other ones. This input would allocate up to all of the buffers. Packets of the previous stage that are directed to the other inputs would be blocked at the previous stage even if their final destination is different from the first packet queued at the hot spot input. Only the hot spot input would contribute to the switch traffic and all other inputs would remain idle.

Additionally, the following assumptions hold for the presented simulation model. However, most of these assumptions can be changed to further interesting

network realizations with little effort due to object oriented modeling. This can be performed by replacing or subclassing the desired components of the object oriented simulation model. The network structure is described by one class while the routing, queuing and various other behaviors of the model are encapsulated in different further classes. So the effort to change the network is minimized. The assumptions of the presented model are:

- All packets have the same size (like in ATM).
- Their destination outputs are distributed uniformly. That means every output of the network is with equal probability one of the destinations of a packet.
- Conflicts between packets are solved randomly with equal probabilities.
- Packets are removed from their destinations immediately upon arrival.
- Routing is performed in pipeline manner. That means the routing process occurs in every stage in parallel.

3 Buffer Structure Comparison

Previously mentioned MINs are simulated for performance evaluation. Networks consisting of switches with shared and non-shared buffers are compared.

The simulation model is implemented in C++ [7]. It handles most kinds of network structures that are based on $c \times c$ switches but is optimized to model MINs. The network is represented as a directed graph starting at the sources (network inputs) and ending at the destinations (network outputs). Packets are generated at the sources. Each packet is provided with a tag determining the destination. Due to multicasting this tag is modeled by a vector of N binary elements, each representing a network output. The elements of the desired outputs are set to “true”. If the packet arrives at a $c \times c$ switch, the tag is divided into c subtags of equal size. Each subtag belongs to one switch output, the first (lower indices) subtag to the first output, etc. If a subtag contains at least one “true” value a copy of the packet is send to the corresponding output containing the subtag as the new tag.

Keeping the amount of allocated memory as small as possible, just a representation of the packets, referred to as containers, are routed along the network paths. The containers are replaced by the packets at the network outputs allowing evaluations. Figure 3 gives a short sketch of the simulation model. So called **ContainerMultiputs** (CM) receive the containers and store them in the queues. At the first network stage, **FirstContainerMultiputs** (FCM) additionally perform the replacement of the packets by containers. So called **ContainerOutputs** (CO) send the containers to the next network stage. At the last stage, **LastContainerOutputs** (LCO) additionally replace the containers by the corresponding packets. Each operation of a switch is aligned by its **Crossbar Manager**. The clocks perform the sequencing of the parallel actions due to computer simulation. The **Deadlock Manager** is just needed in case of multicast and wormhole routing. Such a scenario, which is not subject to this paper, may cause deadlocks.

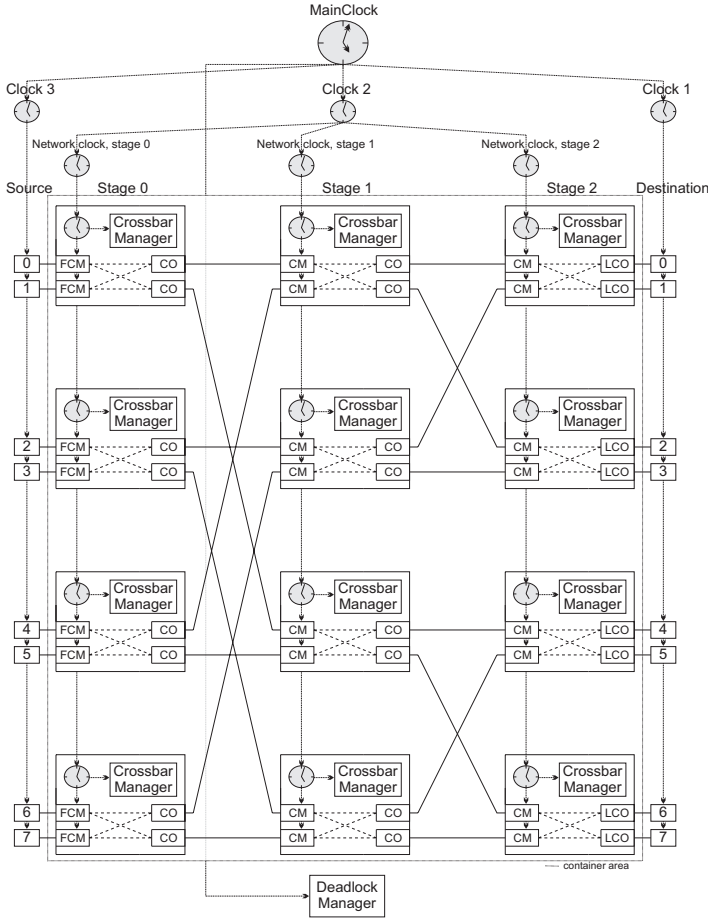


Fig. 3. Sketch of the simulation model

Simulations are performed starting multiple simulation runs in parallel and using a confidence level of 0.95 and a relative error of 0.02 as termination criteria. The simulation is observed and managed by the tool *Akaroa* [5].

All following figures identify the results of non-shared buffer networks by a legend “non-shared buffer: x , total z ” where x represents the buffer size of each FIFO buffer and $z = c \cdot x$ gives the overall buffer size of the switch. Shared buffer networks are identified by a legend “shared buffer: min v , max w , total z ” where v represents the minimal buffer size of each input, w the maximal buffer size of each input, and z gives the overall buffer size of the switch.

The figures show the average throughput and delay times of 16×16 MINs consisting of four stages of 2×2 switches. The packets are routed by cut-through switching.

First, a completely uniform network traffic is investigated: the offered load to all inputs is equal. Concerning multicasting, all output combinations occur with equal probability as the destination of a packet entering the network. Figure 4 shows the dependence between offered load and average throughput at the outputs in case of uniform network traffic. Increasing the offered load from 0.01 to

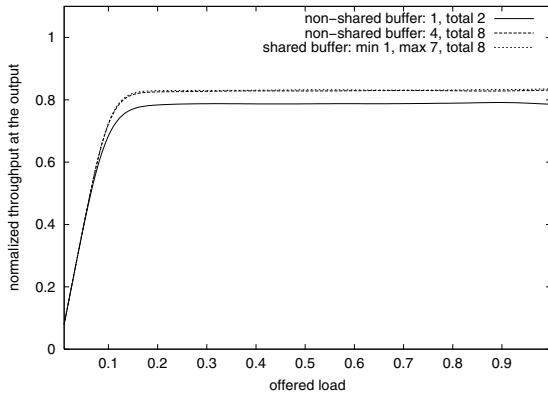


Fig. 4. Uniform network traffic

1.0, the network reaches congestion for an offered load greater than approx. 0.14 due to the large number of packets caused by multicasting.

Comparing the networks with shared and non-shared buffers of an overall buffer size of 8, no observable difference in throughput occurs. In case of uniform network traffic the buffer structure does not affect the throughput. However, larger buffers result in higher throughput.

In the following, uniform traffic is replaced by merging sources sending unicast traffic and sources sending broadcast traffic.

First, traffic established by one broadcast source is investigated. The offered load of this source is varied from 0.01 to 1.0. All other sources send unicast traffic to the network with a fixed offered load of 0.2. Figure 5 shows the throughput of the merged traffic at the outputs for various buffer sizes and structures. Shared buffers perform a higher throughput than non-shared buffers of the same size: buffer space is more efficiently used. On the other hand, a more efficiently used buffer results in larger packet queues at the switch inputs: higher delay times occur (Figure 6).

A further investigated traffic pattern is similar to the previously mentioned one except the fact that two broadcast sources are used. These sources may be located at various network inputs. Depending on their locations the first conflicts between their packets will occur in different network stages. E.g., if they are located at the first two inputs, they are already in conflict at the first network stage because they are located at the same switch. If they are located at

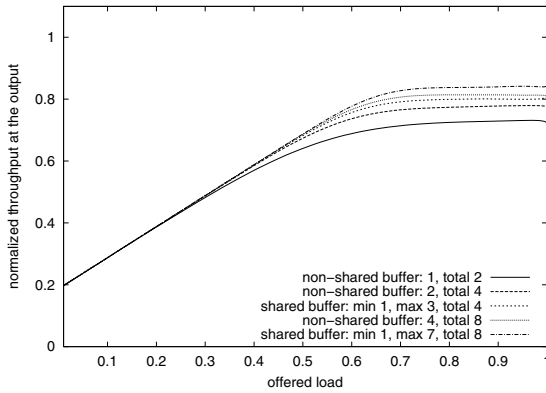


Fig. 5. One broadcast source (throughput)

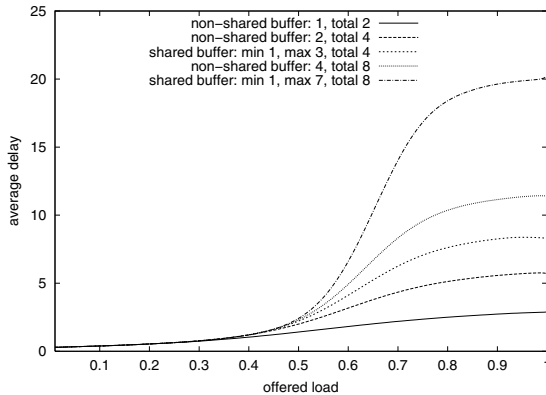


Fig. 6. One broadcast source (delay)

the first and last input, the first crossing of their packets' network paths occurs soonest at the last stage.

Figure 7 shows the throughput taking the stage of the first conflict into account. A non-shared switch input buffer of 1 is chosen. The sooner the network paths of both broadcast sources cross, the lower is the network throughput. E.g., if the crossing and therefore the first conflict occurs at the first stage (stage 0), the output of this switch equals the output in case of just one broadcast source at the inputs: both sources send a packet to both outputs but no more than one packet can pass an output. Just one broadcast source would also result in one packet passing each of both outputs.

If MINs are fed by more than one high-load source the sources should be placed in such a way that their paths are crossing as late as possible.

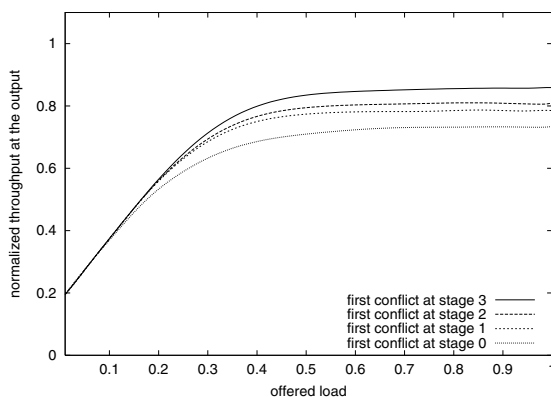


Fig. 7. Two broadcast sources

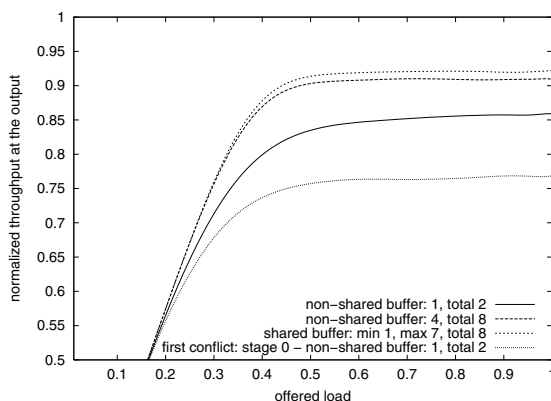


Fig. 8. First conflict at last stage

A comparison of shared and non-shared buffer structures depending on the stage of the first conflict is presented in Figures 8 and 9. Figure 8 demonstrates the throughput behavior for two broadcast sources that cause their first conflict at the last stage. The throughput behavior is given for various buffer structures. Additionally the dotted line allows a comparison to a source distribution resulting in a first conflict at the first stage.

Broadcast sources that cause their first conflict at the first stage are evaluated in Figure 9. This figure also shows the throughput of various buffer structures. The dotted line determines the throughput of a source distribution resulting in a first conflict at the last stage.

All figures show a higher throughput for network switches with a shared buffer compared to switches with non-shared buffers. However, the throughput

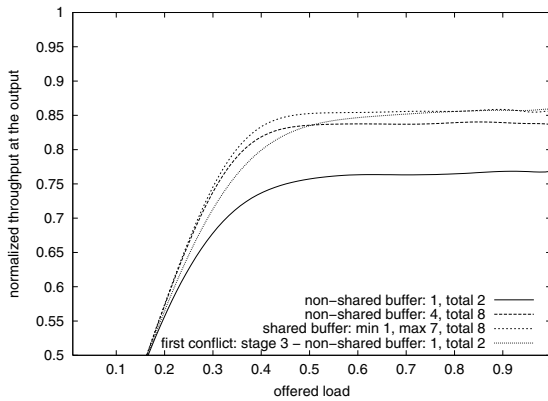


Fig. 9. First conflict at first stage

increases from non-shared to shared buffers by only a small amount and only in case of network congestion. This increase is paid with a more complex switch hardware for managing the shared buffer.

4 Conclusion

Multistage interconnection networks are often proposed to establish a multiprocessor system, ATM switches, or Ethernet switches. Various MIN structures exist to improve the performance. This paper compares the two buffer structures of shared and non-shared buffers in case of packet multicast. Shared buffers perform a dynamic buffer allocation. At least, space for one packet is reserved for each input avoiding the isolation of inputs.

In case of uniform network traffic, both buffer structures show identical behavior. Non-uniform traffic causes a slightly higher network throughput if shared buffers are used. On the other hand, shared buffers require a more complex switch management.

If the network operates at high traffic load, e.g. caused by multicasting, the network stage of the first crossing of high load paths influences heavily the network behavior.

References

1. Gheith A. Abandah and Edward S. Davidson. Modeling the communication performance of the IBM SP2. In *Proceedings of the 10th International Parallel Processing Symposium (IPPS'96)*; Hawaii. IEEE Computer Society Press, 1996.
2. M. Atiquzzaman and M. S. Akhtar. Performance of buffered multistage interconnection networks in a non uniform traffic environment. *Journal of Parallel and Distributed Computing*, 30(1):52–63, October 1995.

3. Ra'ed Y. Awdeh and H. T. Mouftah. Survey of ATM switch architectures. *Computer Networks and ISDN Systems*, 27:1567–1613, 1995.
4. Daniel M. Dias and J. Robert Jump. Analysis and simulation of buffered delta networks. *IEEE Transactions on Computers*, C-30(4):273–282, April 1981.
5. Greg Ewing. *Akaroa II. Version 1.2. User's Manual*. New Zealand, July 1996.
6. Ming-Huang Guo and Ruay-Shiung Chang. Multicast ATM switches: Survey and performance evaluation. *ACM Sigcomm: Computer Communication Review*, 28(2):98–131, April 1998.
7. Matthias Hendler. Simulative Leistungsbewertung von gepufferten mehrstufigen Kommunikationsnetzen bei Cut-Through-Switching (in German). Master's thesis, Technische Universität Berlin, Germany, 2000.
8. Yih-Chyun Jenq. Performance analysis of a packet switch based on single-buffered banyan network. *IEEE Journal on Selected Areas in Communications*, SAC-1(6):1014–1021, December 1983.
9. Jaehyung Park and Hyunsoo Yoon. Cost-effective algorithms for multicast connection in ATM switches based on self-routing multistage networks. *Computer Communications*, 21:54–64, 1998.
10. Wenge Ren, Kai-Yeung Siu, Hiroshi Suzuki, and Masayuki Shinohara. Multipoint-to-multipoint ABR service in ATM. *Computer Networks and ISDN Systems*, 30:1793–1810, 1998.
11. Neeraj K. Sharma. Review of recent shared memory based ATM switches. *Computer Communications*, 22:297–316, 1999.
12. Rajeev Sivaram, Dhableswar K. Panda, and Craig B. Stunkel. Efficient broadcast and multicast on multistage interconnection networks using multiport encoding. *IEEE Transaction on Parallel and Distributed Systems*, 9(10):1004–1028, October 1998.
13. Dietmar Tutsch. Generating systems of equations for performance evaluation of buffered multistage interconnection networks. Technical Report FB Informatik 2000–07, Technische Universität Berlin, 2000.
14. Dietmar Tutsch and Günter Hommel. Multicasting in buffered multistage interconnection networks: an analytical algorithm. In *12th European Simulation Multi-conference: Simulation – Past, Present and Future (ESM'98)*; Manchester, pages 736–740. SCS, June 1998.
15. Dietmar Tutsch and Günter Hommel. Multifractal multicast traffic in multistage interconnection networks. In *Proceedings of the High Performance Computing Symposium 2001 (HPC 2001)*; Seattle, accepted for publication, April 2001.
16. Yuanyuan Yang. An analytical model for the performance of buffered multicast banyan networks. *Computer Communications*, 22:598–607, 1999.
17. Hyunsoo Yoon, Kyungsook Y. Lee, and Ming T. Liu. Performance analysis of multi-buffered packet-switching networks in multiprocessor systems. *IEEE Transactions on Computers*, 39(3):319–327, March 1990.
18. B.Y. Yu. Analysis of a dual-receiver node with high fault tolerance for ultrafast OTDM packet-switched shuffle networks. Technical paper, 3COM, 1998.
19. Bin Zhou and M. Atiquzzaman. Efficient analysis of multistage interconnection networks using finite output-buffered switching elements. *Computer Networks and ISDN Systems*, 28:1809–1829, 1996.

Performance Evaluation of PIM-SM Recovery

Tarik Čičić¹, Stein Gjessing¹, and Øivind Kure²

¹ University of Oslo, Department of Informatics
P.B. 1080 Blindern, 0316 Oslo, Norway
{tarikc, steing}@ifi.uio.no

² Norwegian University of Science and Technology
Center for Technology at Kjeller
P.B. 70, 2027 Kjeller, Norway
okure@unik.no

Abstract. A PIM-SM-built multicast tree must be restructured/recovered when the underlying unicast routing tables change. In this article we describe the PIM-SM recovery mechanisms and evaluate the recovery performance, showing its dependence on a range of network and session parameters. Our results show that a significant recovery performance improvement is possible if the multicast recovery is immediately triggered when the unicast routing state changes. Furthermore, our results show that a substantial packet loss can be caused by non-reductive, “benign” events in the network, such as an addition of a new link.

1 Introduction

Stephen Deerings Ph.D. dissertation and the ensuing work in IETF on multicast protocols were the foundation for IP multicast [1, 2, 3]. The subsequent establishment of Mbone [4] positioned IP multicast as an emerging, powerful IP technology supporting a range of new, primarily multimedia applications. To address the inherent scalability problems of this technology, “Protocol Independent Multicast – Sparse Mode” (PIM-SM, [5]) was developed, and it is the most widely used multicast routing protocol today.

PIM-SM creates and maintains unidirectional multicast trees based on explicit Join/Prune protocol messages. These control messages are sent on a node-to-node basis. PIM is “protocol independent” in the sense that it is independent of the underlying unicast protocol — it can run on top of any unicast routing protocol.

To build a multicast tree, PIM multicast routers use a mechanism called Reverse Path Forwarding [6]. RPF determines the direction to the root of the tree using the unicast routing tables. This information is used to select an interface on which Join/Prune messages are sent, and where the multicast packets originated at the root are expected to arrive. Based on received Join/Prune messages, routers maintain a set of mappings between the input interface and the output interfaces for each known multicast group.

In case of unicast routing change, all multicast routing entries are reexamined using the RPF mechanism in order to determine the (possibly) new input

interface. This process of reestablishing the multicast tree we call *tree recovery*. If the new input interface differs from the old one, the multicast routing entry is updated: the new input interface is set instead of the old one and the new input interface is removed from the output interface list, if it was in it. Finally, control messages are sent to the neighboring routers: Join at the new input interface and Prune at the old input interface, if it is operational. In the transient phase, from the unicast routing change to the stabilization of the new multicast tree, packet loss may occur.

PIM-SM has received substantial attention in the research community [7, 8]. Also, significant research has been done on application-level error recovery for real-time IP multicast [9] and reliable multicast applications [10]. However, there has been less attention on the multicast tree recovery at the network level. Wang et al. [11] focussed on the performance of fault recovery in PIM Dense Mode running over OSPF. In addition, they analyzed the qualitative aspect of fault recovery of PIM running over OSPF. Our work extends these results and focuses on the performance of PIM-SM recovery.

2 Problem Statement

The performance of PIM-SM tree recovery is influenced by a range of factors, including the network topology properties (e.g. average node degree, link delay), multicast session properties (e.g. group size and data flow properties) and routing mechanisms (e.g. unicast routing protocol and multicast recovery initiation method).

In this paper we explore the effect of these parameters on the performance of PIM-SM recovery. In particular, the multicast recovery initiation can be based on periodic polling of the unicast routing tables (*periodic recovery*), or on receiving of an explicit change notification from the unicast routing process (*triggered recovery*). The periodic recovery is more common in practice, since it does not assume that the unicast routing is aware of the multicast routing. In our work we analyze performance and cost aspects of both mechanisms.

The unicast routing changes are caused by events belonging to three broad classes: *Topology Reduction*, e.g. link failure, removal or node failure, *Topology Enrichment*, e.g. link recovery or adding a new link and *Dynamic Routing Change*, e.g. link metric change. If topology reduction has occurred, the packet loss is often inevitable, since it takes time to reconstruct the multicast tree using alternative links. Intuitively, events belonging to the other two classes, called *benign events* in the rest of this paper, should not cause any packet loss. However, the standard PIM-SM recovery procedure implies that, in the case of a changed input interface, the old input interface is immediately disabled. In other words, events such as enrichment of the network by a new, operational link can also cause multicast packet loss. In this paper we evaluate the PIM-SM recovery performance both in the case of topology reduction (link failure) and a benign event (link recovery).

3 Performance Evaluation

We have developed a simulation model of PIM-SM [12] using the Network Simulator (NS) framework [13]. The model provides a general implementation of PIM-SM (routing based on explicit Join/Prune protocol messages, soft state with periodic refresh etc.) and a detailed implementation of the PIM-SM recovery [5]. The model is parameterized through a range of parameters including the average node degree, link delay, group density, CBR source rate etc. The unicast routing is based on the NS's standard distributed implementation of the Distance Vector protocol. We use random network topologies constructed to reflect real transport networks [14, 15].

In each simulation instance, after the multicast distribution tree has stabilized and the source has started to send data, a randomly chosen link within the multicast tree is taken down. This event we call “link-down” event. After the multicast tree has recovered, the link is reintroduced in the network (“link-up” event). We measure the packet loss in receivers caused by these events.

To evaluate the effect of the different parameters on PIM-SM recovery performance, we conduct a set of simulations where the parameters are varied within anticipated real network values. The following parameter ranges are chosen: recovery mechanism (periodic $p=20\text{ms}$, periodic $p=50\text{ms}$ or triggered) average node degree ($D=\{2.5, 3.0, \dots, 5.0\}$) and group density (5, 10, 15, 20 receiver nodes out of 30 in the network). The average link delay in all test networks is 3ms, bandwidth 10Mb/s, CBR rate is 500packets/second and the packet length is 320Bytes.

3.1 Performance Evaluation Basis

In this subsection we first analyze unicast recovery. We find the average packet loss in a unicast data flow when a link goes down, under the same conditions as in the forthcoming multicast study. We will use these results as a comparison for the multicast recovery performance. Furthermore, we present how many multicast receivers are affected by the tree recovery and how often the packet loss occurs. These data are significant for a proper evaluation of the multicast packet loss figures presented later in this section.

We believe that it is only of interest to consider simulation instances where tree recovery after link-down is possible. Hence, we are not considering simulation instances where the link-down event resulted in disconnected topology.

Unicast Loss. Our study is performed in networks using a unicast routing protocol based on the Distance Vector (DV) algorithm. When DV is used, each node has sufficient information to immediately repair the failed route if and only if the alternative route is two hops long. If three or more hops are necessary, the upstream node will discard the packets while the routing updates are exchanged and the routing state converges. This period will be longer in sparse networks due to longer alternative routes.

If a Link State (LS) protocol implementation is used, the unicast packet loss is not dependent on the average node degree. The unicast routing update flooding starts as soon as the link failure is detected. After receiving the update, each node can calculate the alternative route instantly. Therefore the packet loss will occur mainly due to the loss of packets traversing the faulty link, and it will be lower than the corresponding one for a DV routing protocol.

We have measured the DV unicast recovery performance as the packet loss in a unicast flow with same properties as the tested multicast flow (10Mb/s CBR flow, 500packets/second, 320Bytes packets). The unicast flow traverses the same faulty link as the multicast flow presented in the remainder of this section.

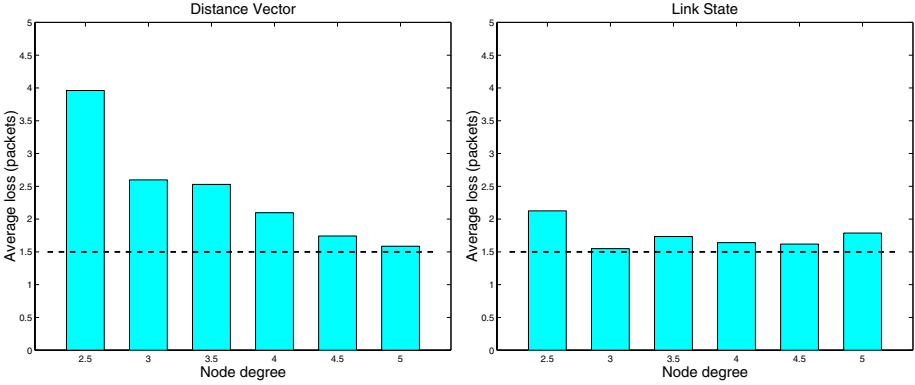


Fig. 1. Unicast packet loss, depending on the average node degree. The expected on-link loss is 1.5 packets because NS excludes the link transmission time in its loss model

Figure 1 (left) shows the Distance Vector unicast loss depending on average node degree. Our results provide a good illustration of the quick recovery in highly connected networks. In 30-nodes networks with the average node degree of 5, the probability of having a two-hop alternative path for a link failure is very high. Hence, the DV packet loss for degree 5 is expected to be just above the minimum, estimated loss of the packets traversing the link:

$$L_{\min} = R \cdot (d_p + d_t) = 500 \text{ s}^{-1} \cdot 0.003256 \text{ s} = 1.628 \quad (1)$$

where R is the packet rate, d_p is the 3 ms link propagation delay and d_t is the 0.256 ms packet transmission time. In our simulation environment the minimum loss is even lower than the estimated minimum (1), since the NS loss model implementation excludes the packet transmission time.

As expected, the LS packet loss is independent of the average node degree (Fig. 1, right).

Affected Receivers. A link failure within the multicast tree will always affect at least one receiver. It is important to present how many receivers are affected in order to gain a complete view of the recovery performance.

The multicast trees are higher (more hops on average from the source to each of the receivers) in the networks with low connectivity than in the networks with high connectivity. Therefore, a link failure is more likely to affect a receiver in a network with low connectivity than in a network with high connectivity.

Figure 2 (left) shows the average number of receivers affected by a single failure. For example, 33% receivers are affected in networks with the average node degree $D=2.5$ and with group size 5, and only 10% in $D=5.0$ networks with 20 receivers.

Sources for Packet Loss. Packet loss may occur due to both link-down and link-up events. The link-down event causes loss in 95% cases in the triggered recovery and almost always in the periodic recovery, regardless of the network parameters. The link-up event, however, causes packet loss only if the input interface has changed, which is more probable in low connectivity networks. This is the case since the alternative paths in the high connectivity networks will in general be shorter and “closer” to the original path — achievable through change of output interfaces in the transit nodes only.

The link-up event causes loss from $\sim 50\%$ cases in $D=5$ networks to 80% cases in $D=2.5$ networks (Fig. 2, right).

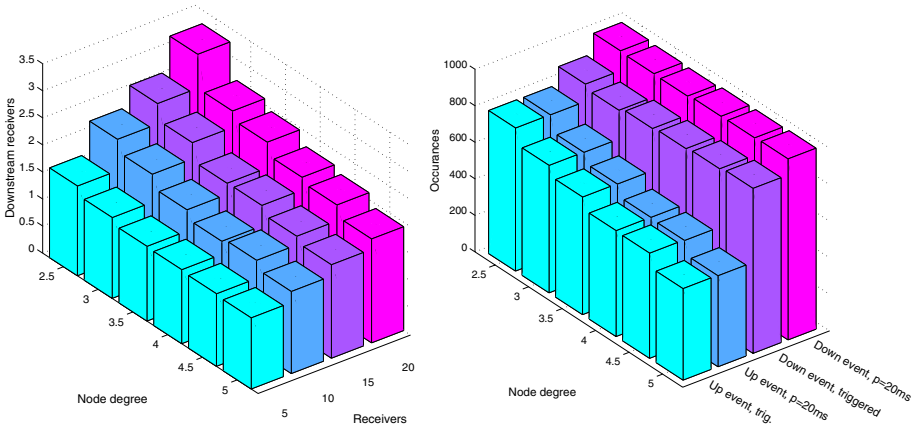


Fig. 2. Consequences of tree recovery: mean number affected receivers (left) and events causing the packet loss out of 1000 simulation instances (right)

3.2 Link-Down Event

When a branch is removed from the multicast tree, the downstream nodes will be cut off until an alternative route is established. The total cutoff time T is bounded:

$$T_m \leq T \leq T_u + p + T_m$$

where T_u is the time it takes to recover the unicast routing, p is the unicast routing check period (20ms and 50ms in our simulations, zero for the triggered recovery) and T_m is the multicast routing recovery time (upstream propagation of the Join-messages to the closest node in the tree).

T_u is shorter for larger average node degrees. T_m decreases as the probability of a nearby in-tree node increases, influenced by the number of receivers and the average node degree.

We expect T to perform much better on average than the worst case. First, the expected time before the multicast recovery starts is $p/2$. Also, the unicast recovery time is often included in this period. Furthermore, the multicast recovery may start before the unicast is completely recovered in the triggered recovery. This happens because the unicast routing recovery takes several routing message exchanges to stabilize, and that the multicast recovery succeeds quickly since the neighboring node may be a member of the same multicast group. In the process of unicast routing, the multicast input interface may temporarily point in wrong direction. This has no effect on the final multicast routing entry, since it is always coherent with the unicast routing.

Our performance evaluation results are shown in Fig. 3. For the same recovery mechanism and number of receivers, each sextuple of adjacent bars represents the six average network degrees (2.5 to 5) we have tested. The standard deviation in these measurements ranges from ~ 2.5 packets for the triggered recovery with 20 group members to ~ 7.5 packets for the periodic recovery with 5 group members.

The unicast loss pattern (Fig. 1, left) is recognizable in the charts for low group sizes. For higher group sizes, the multicast recovery often succeeds before the unicast is completely recovered due to the high probability of the neighbor node being a member of the multicast group, thereby obscuring the unicast loss pattern.

The effect of the node degree and the group size is shown in the characteristic pattern where the performance increases by ~ 3 packets from $D=2.5$ networks with 5 receivers to $D=5.0$ networks with 20 receivers, for both periodic and triggered recovery.

We can observe that the loss performance is dominated by the unicast routing check period p : the mean loss value for the triggered, periodic $p=20$ ms and periodic $p=50$ ms recovery is 4.5, 8.8 and 15.8 packets, respectively. The difference between the first two is 4.3 packets. The expected time between the link down event and the recovery procedure initiation is $p/2=10$ ms or 5 packets. The 0.7 packet difference is caused by the overlap between the unicast and multicast recovery.

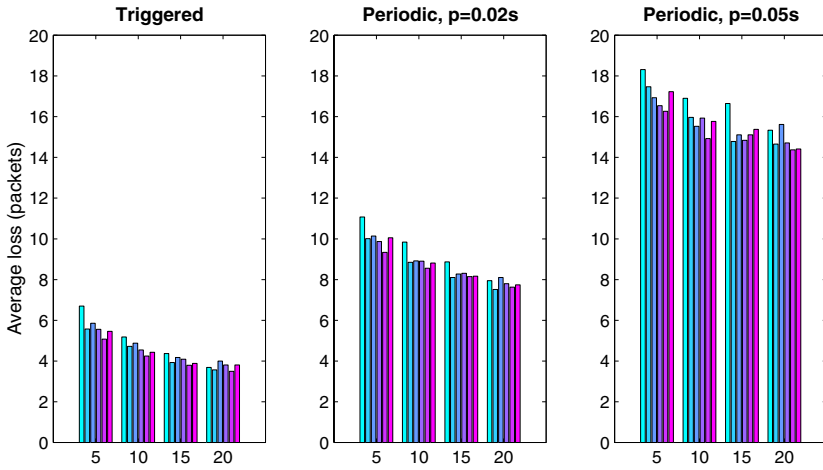


Fig. 3. Mean packet loss per affected receiver, link-down event. Each sextuple of adjacent bars represents the six average node degrees: 2.5 (leftmost) to 5 (rightmost) links per node, for group sizes 5, 10, 15 and 20 receivers

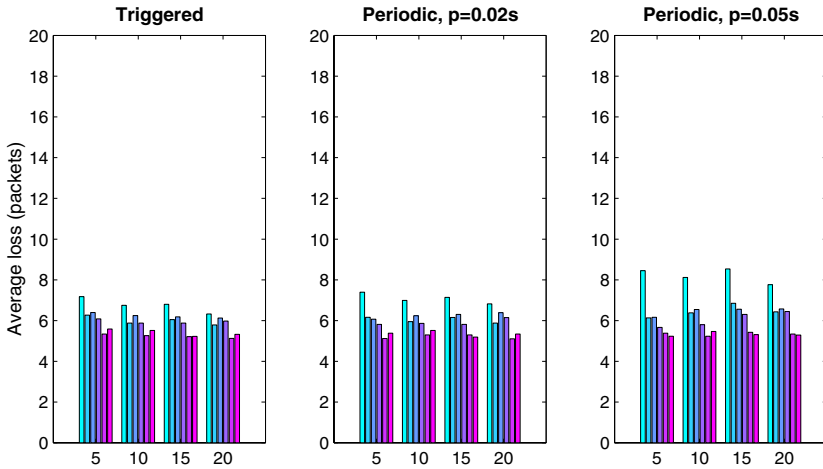


Fig. 4. Mean packet loss per affected receiver, link-up event. Each sextuple of adjacent bars represents the six average node degrees: 2.5 (leftmost) to 5 (rightmost) links per node, for group sizes 5, 10, 15 and 20 receivers

3.3 Link-Up Event

When a network link recovers, the unicast routing tables are updated for routers that have the link in a shortest path route. In our scenario, the unicast routing tables become the same as before the link-down event. The multicast routing process notices this benign event, and starts the recovery procedure in order to reestablish the better multicast tree.

The PIM-SM recovery procedure implies that the old input interface on a router is closed instantaneously when the new input interface is chosen. It takes time for the multicast flow to propagate over the new branch. The packet loss in this case is dependent on the branch propagation delay, which has fewer hops and a shorter delay in networks with high node degree.

The mean packet loss caused by the link-up event is shown in Fig. 4. The packet loss is largely independent of the recovery period, since the old input interfaces are operational and unchanged even though the unicast routing changes in this period.

4 Overhead Comparison

PIM-SM tree recovery includes the multicast routing table recalculation and the exchange of Join/Prune control messages on the new links. These actions will respectively cause additional router CPU consumption and the network load increase. We provide an estimate of how often the overhead is incurred for the the two recovery types (periodic/triggered).

Computational Overhead. Each time the unicast routing state has changed, an RPF check has to be done for each multicast routing entry. The triggered recovery will be invoked only when the changes have occurred, justifying the routing table processing. If Distance Vector unicast routing is used, the procedure can be invoked repeatedly as the unicast routing stabilizes. This will additionally stress the system in the transient phase.

The periodic recovery needs to know if there has been any changes in unicast routing tables since the last invocation of the recovery procedure. If this is impossible, because e.g. the unicast routing is unaware of the coexisting multicast routing and does not release the last update information, the only implementable solution is the least effective, periodic recovery with unicast routing table processing in each invocation.

Communication Overhead. In PIM-SM recovery the communication overhead consists of two parts, the transmission of packets that are rejected due to wrong input interface and the Join/Prune messages triggered by multicast routing changes. The PIM-SM standard specifies sending the prune message on the old input interface and merging Join/Prune messages for many multicast groups, thereby minimizing the communication overhead.

The periodic recovery will send at most one Join/Prune message, and only if the input interface has changed. The triggered recovery may send several Join/Prune messages, as the DV routing stabilizes.

Repeated Recovery Invocations in Triggered Recovery. To understand the amount of the additional overhead in transient period in the triggered recovery, we have counted the number of calls and the number of input interface changes in our simulator. We have tested D=3.0 topologies with a single, five-member group.

The link-down event caused an average of 1.75 recovery procedure invocations per multicast node. A maximum of 13 invocations was registered, however, 4 or fewer invocations were registered in more than 95% cases. A maximum of 5 Join/Prune messages per node was sent. In 75% cases the input interface remained the same (zero Join/Prune messages sent), and in additional 20% cases a single Join/Prune was sent.

This result shows that, in our DV simulation environment, the triggered recovery induced a 75% higher computational overhead and a slightly higher control message overhead, as compared to the periodic recovery.

Link State Routing. A Link State unicast routing protocol will receive at most one unicast routing update for each event (e.g. single link removal). This implies that at most one multicast recovery procedure invocation will occur, even in the triggered recovery. In other words, in LS-based networks, the triggered and the periodic recovery will have similar computational and communication overhead.

5 Conclusion

We have evaluated the PIM-SM recovery performance depending on the recovery mechanism and various topology and session parameters. Packet loss occurs due to both reductive and benign events. We simulated a reductive event as a link failure (link-down event) and a benign event as the link recovery (link-up event).

The link-down event causes packet loss in at least 95% cases in our test environment, regardless of the other parameter settings. The triggered recovery has superior performance as compared to the periodic recovery. The triggered recovery will in general have computational and communication overhead of the same order as the periodic recovery, but may not be implementable on some systems. Other factors (e.g. average node degree) have a moderate effect on the performance. In general, PIM-SM recovers quickly, showing performance close to the underlying unicast recovery.

The packet loss caused by the link-up event is unnecessary high, and can be decreased using an improved recovery algorithm. Detailed specification and analyze of this algorithm is the topic of our current research.

References

1. Steve Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, 1991.
2. David Waitzman, Craig Partridge, and Steve Deering. Distance vector multicast routing protocol. RFC 1075, November 1988.
3. Steve Deering. Host extensions for IP multicasting. RFC 1112, August 1989.
4. Hans Eriksson. MBONE: The multicast backbone. *Communications of the ACM*, 37(8), August 1994.
5. Deborah Estrin, Dino Farinacci, Ahmed Helmy, David Thaler, Stephen Deering, Mark Handley, Van Jacobson, Ching gung Liu, Puneet Sharma, and Liming Wei. Protocol independent multicast – sparse mode (PIM-SM): Protocol specification. RFC 2362, June 1998.
6. Yogen K. Dalal and Robert M. Metcalfe. Reverse path forwarding of broadcast packets. *Communications ACM*, 21:1040–1048, December 1978.
7. Liming Wei and Deborah Estrin. Multicast routing in dense and sparse modes: simulation study of tradeoffs and dynamics. In *Computer Communications and Networks, Fourth International Conference on*, pages 150–157. IEEE, September 1995.
8. Tom Billhartz, J. Bibb Cain, Ellen Farrey-Goudreau, Doug Fieg, and Stephen Gordon Batsell. Performance and resource cost comparison for the CBT and PIM multicast routing protocols. *IEEE J. Selec. Areas Commun.*, 15(3):304–315, April 1997.
9. Georg Carle and Ernst W. Biersack. Survey of error recovery techniques for IP-based audio-visual multicast applications. *IEEE Network*, pages 24–36, November 1997.
10. Christophe Diot, Walid Dabbous, and Jon Crowcroft. Multipoint communication: A survey of protocols, functions, and mechanisms. *IEEE J. Selec. Areas Commun.*, 15(3):277–290, April 1997.
11. Xin Wang, C. Yu, Henning Schulzrinne, Paul Stirpe, and Wei Wu. IP multicast fault recovery in PIM over OSPF. In *Proceedings ACM SIGMETRICS*, June 2000.
12. Tarik Čičić, Stein Gjessing, and Øivind Kure. Tree recovery in PIM sparse mode. Research Report 293, University of Oslo, Department of Informatics, March 2001. ISBN 82-7368-243-9.
13. UCB/LBNL/VINT. Network simulator - ns (version 2). WWW. [HTTP://WWW.ISI.EDU/NSNAM/NS/](http://www.isi.edu/nsnam/ns/).
14. Bernard M. Waxman. Routing of multipoint connections. *IEEE J. Selec. Areas Commun.*, 6(9):1617–1622, December 1988.
15. Ellen W. Zegura. Georgia tech internetwork topology models. WWW. [HTTP://WWW.CC.GATECH.EDU/FAC/ ELLEN.ZEGURA/GRAPHS.HTML](http://www.cc.gatech.edu/fac/ellen.zegura/graphs.html).

Reducing Multicast Inter-receiver Delay Jitter – A Server Based Approach

Jens-Uwe Klöcking, Christian Maihöfer, and Kurt Rothermel

Institute of Parallel and Distributed High-Performance Systems (IPVR),
University of Stuttgart, Breitwiesenstr. 20-22, 70565 Stuttgart, Germany
`Jens.Kloecking@informatik.uni-stuttgart.de`

Abstract. We consider the problem of how to achieve a simultaneous arrival of information at a multitude of recipients for applications where the receivers are non-cooperative. For that reason, we aim at designing an inter-receiver delay jitter fair service for Internet multicast delivery. In contrast to related work, we present an approach at application layer, which does not assume special properties of the core network nodes and can be partially deployed. Additionally, it implicitly takes current network load into account, which gives the opportunity to keep the overall message delivery delay low. Simulation results show the feasibility of the described algorithms and a significant reduction of inter-receiver delay jitter compared to normal multicast delivery.

1 Introduction

Multicasting allows an efficient transmission of a message to a group of receivers. A multitude of new application areas are based on this transmission technique, e.g. news and software distribution, distributed computing and multimedia applications like videoconferencing and teleteaching. Combined with current multicast transport protocols, multicasting provides a suitable distribution platform for information. However, as information becomes more and more valuable and therefore instant reaction to information is necessary, subscribers to information services would like to be assured that the provided information is not delayed with respect to other recipients. For example, assume a system that provides sensitive stock quote information. The sooner a recipient gets this information the more money he is likely to earn. Especially if the receivers have to pay for this channel they do not tolerate delay unfairness. Hence, a service is required that achieves fairness regarding the delay jitter among receivers, which is not given in the current Internet. We will call this property delay-fairness for short. Besides information services, other applications like electronic markets and distributed games would benefit from a delay-fair service, too.

The approach presented in this paper improves delay fairness by using servers in the network to smooth message reception time differences at multicast receivers. Our approach is based on a realistic system model of the current Internet. Though it is not possible without reservation of bandwidth and without router

support to guarantee absolute delay fairness, our simulation results show a significant decrease of inter-receiver delay jitter. As our approach is completely at application layer and therefore needs no router modifications it can be deployed partially and in short time.

The paper is structured as follows. First, we define the problem and discuss related work. In Section 3 we present our server-based approach to improve delay fairness. Simulation results are presented in subsection 3.4. Finally, section 4 concludes the paper with a brief summary.

2 Problem Definition and Related Work

We assume a sender s and a set of receivers $R = \{r_1, \dots, r_j, \dots, r_n\}$ of a multicast group. The sender sends messages $m_1, \dots, m_i, \dots$ to that multicast group. The delay that is experienced by message i on its way from Sender s to Receiver r_j is denoted by D_{ij} . The inter-receiver delay jitter of message i is the difference between the first reception of message i at a receiver and the last reception of the same message at another receiver:

$$J_i = \max\{\forall r_j \in R : D_{ij}\} - \min\{\forall r_j \in R : D_{ij}\}$$

We define the delivery delay for message i as

$$D_i = \max\{\forall r_j \in R : D_{ij}\}$$

These definitions are illustrated in Figure 1. We aim at minimizing the inter-receiver delay jitter while keeping the delivery delay in reasonable bounds.

Whereas we focus in this paper on an approach at application layer, related work concentrates on network aspects. The consideration of inter-destination delay jitter during the multicast routing tree set up is in the main interest [11,5,13,12]. An approach to compensate the jitter caused by the network is presented in [4]. The algorithm is based on a delay-fair multicast tree and additionally assumes that the packet scheduling delay is bound and known. The nodes are modelled as a regulator followed by a packet scheduler for each outgoing link. The regulator delays each packet until its eligible time to be scheduled is reached. The calculations of the algorithm are based on worst-case assumptions of the packet scheduler delay bounds. The current network load is not considered.

Our work differs from related work in the following way. We propose an approach that works at application layer and is therefore deployable in short time. Our main goal is to present a scalable approach which fits well to the current Internet structure without the need for router changes.

3 A Server Based Architecture for Inter-receiver Delay Jitter Smoothing

The applications that can benefit from an inter-receiver delay fair service differ widely in their data rate requirements. At the moment we focus on applications

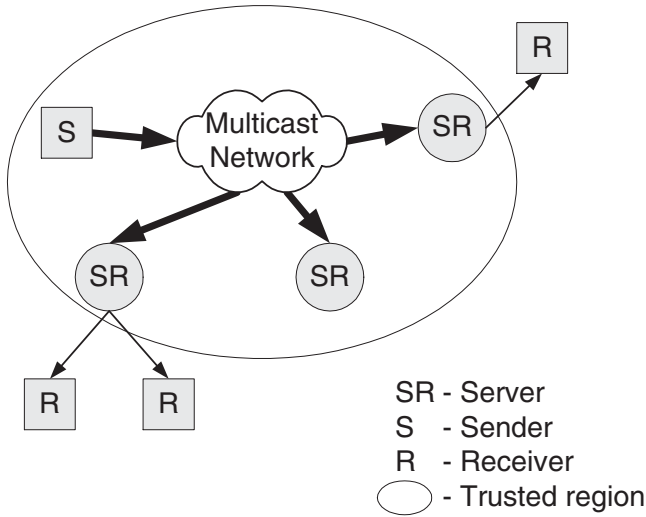


Fig. 2. Placing servers in the trusted region

The synchronisation of the data messages at the servers will result in an inter-receiver delay jitter smoothing. However, as we assume a packet data network without synchronous message delivery there is no way to tell all servers at the same time that they can instantly reveal the information. Besides, this is actually the problem we are about to solve. This means that the sender has to predict the time at which the information will have been arrived at all servers and to send this predicted time along with the message. We will call this time information revealing time. The clocks of the information source and the servers are assumed to be synchronized. For example the Network Time Protocol NTP [8] achieves synchronisation accuracies in the order of tens of milliseconds over Internet paths. The server based inter-receiver delay jitter smoothing is depicted in Figure 3. Basically our approach works in the following way:

1. Determine an information revealing time at the sender. This can be done by estimation, configuration or by a test message that contains no application data.
2. The sender transmits the data message including the revealing time with multicast to the set of servers.
3. The servers receive the data and deliver it at the revealing time to the connected receivers. This can be achieved by unicast or by a server specific multicast group. The server provides feedback information about the suitability of the information revealing time to the sender (see section 3.3 for details).
4. The receivers have to deliver the incoming data from the servers instantly to the application.
5. The sender collects the feedback information from the servers.

6. The sender determines a new revealing time based on the received feedback.
7. Proceed with step 2.

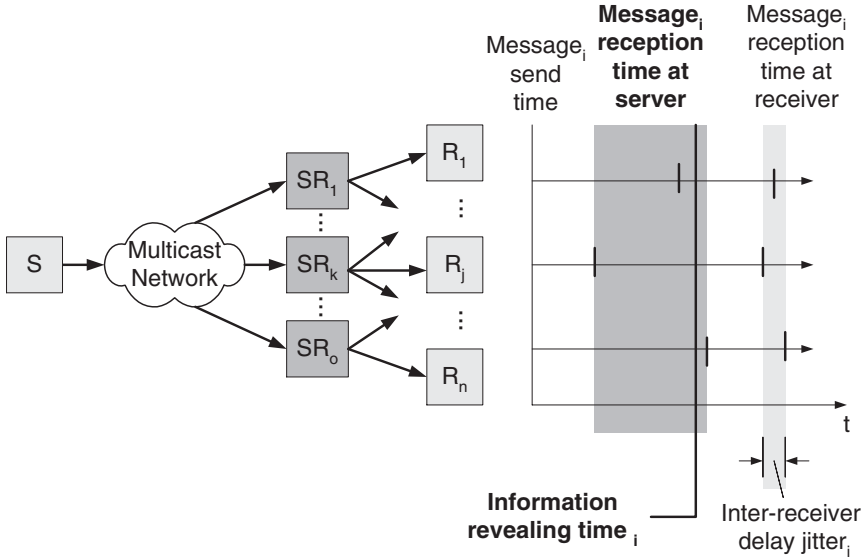


Fig. 3. The servers forward the messages at the information revealing time specified by the sender to smooth the inter-receiver delay jitter

From the servers the data is delivered with multicast or unicast to the receivers depending on their number attached to one server. Note that the delay between the servers and the receivers cannot be taken into account when delivering the message. The reason is that it is not possible for the servers to determine the delay to the untrusted receivers in a secure way. For example, if a server tries to measure the delay by sending ping requests to the receivers, the answer can be deliberately delayed, which would result in an earlier delivery of the messages to that receiver.

The following sections will describe the feedback mechanism and the algorithm to determine the revealing time in more detail.

3.2 Feedback Protocol

Servers have to provide feedback to the sender to allow a dynamic reaction to changing network conditions. A simple approach would be to send the feedback with unicast. However, we have learned from reliable multicast implementations that such an approach may result in a large number of messages, which leads to an overwhelming of the sender and therefore limits scalability [7]. Our approach is based on the idea of suppressing the feedback information, which is also used in the more scalable reliable multicast protocols.

Alternative 1: Revealing Time Expiration Notification. The servers send a multicast feedback message if the information revealing time has already expired at data message reception time. The feedback message contains only the message sequence number. This tells the sender to increase the revealing time without transmitting an exact value. To accomplish that not all servers answer at the same time, the feedback send time is chosen randomly. The feedback is suppressed either if one of the servers has answered to the same message sequence number already or if not at least a message round trip time has passed since the last sent feedback information. The feedback tells the source that the deadline was too short for at least one server and therefore should be extended. Since the feedback is only gathered from the servers where the data has arrived too late, there is no feedback information (and thus less message overhead) necessary in the case where all servers got the data in time. Therefore, an algorithm like the multiplicative increase, additive decrease algorithm (see Section 3.3) is necessary to find a suitable deadline value.

Alternative 2: Desired Revealing Time Notification. A further option is to send the feedback with the message sequence number and the desired future delivery delay included. The desired delivery delay is estimated with regard to the mean and variation of previous received messages D_{ij} using formulas established by Jacobson [6] for TCP round trip time calculation. A server can request an extended or reduced information revealing time. Though, the sender is interested only in the highest information revealing time. This requires another suppression mechanism. Again the feedback is sent to the multicast group to which all servers belong. A server sends feedback scheduled at a random time provided that no feedback has been received with a higher or equal information revealing time for the same message sequence number. The delay D_i can be determined with the filter algorithm discussed in section 3.3.

There are further design options for the feedback protocol that we will not describe here in more detail since they are based on more extensive requirements. Besides sending the feedback with multicast to the sender and the other servers, the feedback could be gathered by a tree-based mechanism. The advantage would be to prevent other servers from receiving feedback information. Such a gathering tree could for example be provided by a Concast service [3].

3.3 Information Revealing Time Calculation

As mentioned, the information revealing time has to be predicted. As the inter-receiver delay jitter is changing over time, a dynamic mechanism is necessary to adapt to network load changes. In this subsection we give for each feedback protocol alternative an example of calculating the information revealing time.

Filter Mechanism. A suitable way to predict the information revealing time is to use feedback information of the message delay (preferred information revealing time), which can be provided by the servers. Using that feedback a new value for the deadline is calculated via a filter operation of the form

$$d_t = a * d_{t-1} + (1 - a) * d_{fdb}, 0 < a \leq 1$$

where d_t is the predicted delay value, d_{t-1} is the predicted delay value of the previous interval, d_{fdb} is the delay value feedback of the previous interval and a is a parameter which adjusts the influence of the measured value to the predicted delay. We transmit the predicted information revealing time along with the data. In this case feedback has to be gathered both if the deadline was sufficiently large and if the deadline was not sufficient.

Increase/Decrease Mechanism. Another option is to predict the information revealing time via an algorithm that multiplicatively increases the deadline if feedback information has been received and additively decreases the deadline if no feedback has been received:

$$d_t = c_1 * d_{t-1}, c_1 > 0$$

$$d_t = d_{t-1} - c_2, c_2 > 0$$

where c_1 and c_2 are constants. Then, no feedback information is necessary in the case where all receivers got the data in time.

3.4 Simulation Results

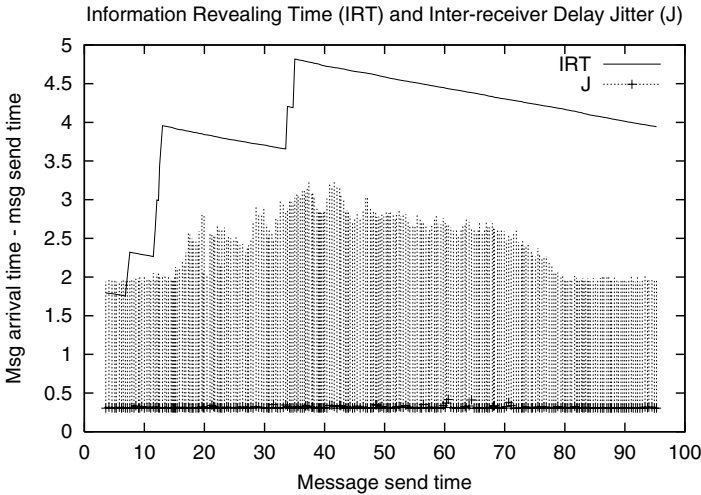


Fig. 4. Revealing time expiration notification protocol and increase/decrease algorithm

We have simulated the multicast distribution of messages sent from an information source to a set of servers to see whether a reduction of the inter-receiver

delay jitter can be achieved and to study the proposed feedback protocols. In the experiments we have analysed to which degree the algorithms minimize the inter-receiver delay jitter and which costs are involved. The costs include the increase of the delay and the message overhead introduced by the algorithm.

The simulations were realized with the Network Simulator NS [1]. The topology generator GT-ITM [14] was used to generate transit stub networks. To account for the jitter due to packet loss we used the multicast transport protocol SRM, though our approach does not depend on a special transport protocol. The simulations do not consider the influence of unsynchronised clocks of the sender and the servers nor the message overhead introduced due to message encryption. We have varied the number of nodes between 10 and 1000 nodes and the sending rate between 1 and 128 kbit/s.

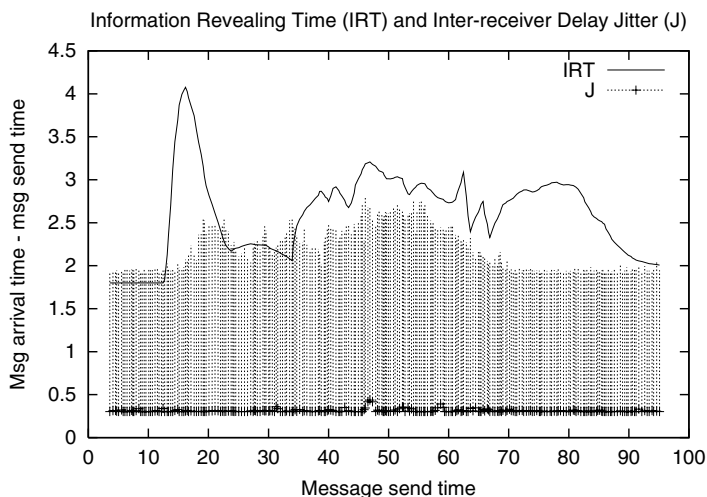


Fig. 5. Desired revealing time notification protocol and filter algorithm

Typical results are shown in Figures 4 and 5 for both information revealing time calculation algorithms. The (dotted) bars in the graph indicate the time span between the first arrival of message i at a server and the last arrival of message i at another server, i.e. the inter-receiver delay jitter at the servers. The information revealing time for each message is shown by the solid line. The figures show the results for a simulation of a 200-node network. One node of the network is designated the information source and 70 nodes of the network are servers. To simulate network load 100 web clients request HTTP traffic at 10 web servers. Additionally, 50 nodes are configured as FTP servers. The FTP traffic is generated between 10 and 70 s simulation time. The data rate of the information source is 32 kBit/s CBR data stream.

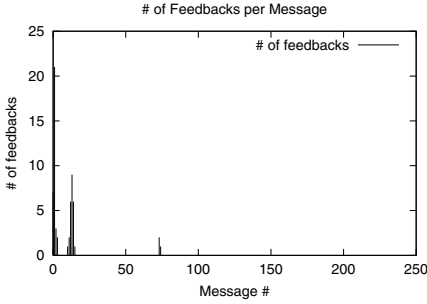


Fig. 6. Feedback overhead for increase/decrease algorithm

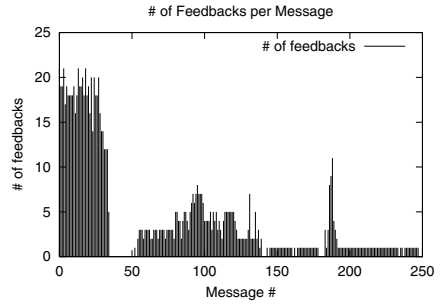


Fig. 7. Feedback overhead for filter algorithm

Whereas the filter algorithm can adapt faster to changing jitter, the increase/decrease algorithm causes less message overhead. With the more conservative increase/decrease algorithm a lower inter-receiver delay jitter is achieved, though, the overall delay is higher. The message overhead of incoming feedbacks at the information source based on the number of outgoing data messages is about 400 percent with the Desired Revealing Time Notification and about 10 percent with the Revealing Time Expiration Notification feedback protocol.

4 Conclusion

In this paper, we examined the multicast fairness problem concerning inter-destination delay jitter. We proposed an approach at application layer. The proposed algorithms make fewer assumptions about the core network nodes and are therefore easier to deploy in the Internet compared to previous work. Furthermore, these algorithms are able to consider current network load, which improves overall message delivery delay. Due to the IP multicast model they need security mechanisms, though. The simulations of the approach showed a significant reduction of inter-receiver delay jitter and a dynamic reaction to network load changes.

At the moment the reactions of the algorithms are coupled with the data rate of the information source. We intend to decouple the algorithms to allow a broader range of applications to use the service. For low data rate applications, test messages can trigger feedback from the servers. Although we intend to develop a transport protocol independent approach, we will examine how efficient a tighter coupling with an appropriate transport protocol would be. For example, the transport protocol control messages could be used to transmit feedback information to optimize bandwidth consumption.

References

1. S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, and S. Kumar. Improving simulation for network research. Technical Report 99-702, University of Southern California, 1999.
2. D. Boggs. Internet broadcasting. Technical Report CSL-83-3, XEROX Palo Alto Research Center, 1983.
3. K. Calvert and J. Griffioen. Internet concast service. Internet Draft, draft-calvert-concast-svc-00.txt, Jul 2000.
4. Ye Ge, Jenifer C. Hou, and Hung-Ying Tyan. A packet eligible time calculation mechanism for providing temporal QoS. In *IEEE International Conference on Communications, Vancouver, BC, Canada*, volume 2, pages 721–6, Jun 1999.
5. Brian K. Haberman and George N. Rouskas. Cost, delay, and delay variation conscious multicast routing. Technical Report TR-97-03, Department of Computer Science, North Carolina State University, Raleigh, NC, USA, Mar 1996.
6. Van Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM '88*, 1988.
7. B. Levine and J. Garcia-Luna-Aceves. A comparison of reliable multicast protocols. *ACM Multimedia Systems*, 6(5):334–348, Sep 1998.
8. D. Mills. Improved algorithms for synchronizing computer network clocks. *IEEE Transactions Networks*, pages 245–54, Jun 1995.
9. Matthew J. Moyer, Josyula R. Rao, and Pankaj Rohatgi. A survey of security issues in multicast communications. *IEEE Network*, 13(6):12–23, Nov 1999.
10. K. Rothermel and C. Maihöfer. A robust and efficient mechanism for constructing multicast acknowledgment trees. In *Proceedings of the Eight International Conference on Computer Communications and Networks (IEEE ICCCN)*, pages 139–45, 1999.
11. G.N. Rouskas and I. Baldine. Multicast routing with end-to-end delay and delay variation constraints. *IEEE JSAC*, 15:346–56, Apr 1997.
12. Quan Sun and H. Langendoerfer. Computation of constrained multicast trees using a genetic algorithm. *European Transactions on Telecommunications & Related Technologies*, 10(5):513–16, Sep 1999.
13. Sun Wensheng and Liu Zemin. Routing multipoint connections in packet-switched computer networks. *High Technology Letters, China*, 5(1):21–4, Jun 1999.
14. E. Zegura, K. Calvert, and M. Donahoo. A quantitative comparison of graph-based models for internet topology. *IEEE/ACM Transactions on Networking*, 5(6):770–783, Dec 1997.

Multicast Routing and Wavelength Assignment in Multi-Hop Optical Networks^{*}

Ran Libeskind-Hadas¹ and Rami Melhem²

¹ Dept. of Computer Science, Harvey Mudd College, Claremont CA 91711, USA
hadas@cs.hmc.edu

² Dept. of Computer Science, University of Pittsburgh, Pittsburgh PA 15260, USA
melhem@cs.pitt.edu

Abstract. This paper addresses multicast routing in multi-hop optical networks employing wavelength-division multiplexing (WDM). We consider a model in which multicast communication requests are made and released dynamically over time. A multicast connection is realized by constructing a multicast tree which distributes the message from the source node to all destination nodes such that the wavelengths used on each link and the receivers and transmitters used at each node are not used by existing circuits. We show that although the *routing and wavelength assignment* in this model is NP-complete, the *wavelength assignment* problem can be solved in linear time.

1 Introduction

Wavelength-division multiplexing (WDM) is emerging as a key technology in communication networks. In WDM networks the fiber bandwidth is partitioned into multiple data channels which may be transmitted simultaneously on different wavelengths. Thus, WDM permits use of enormous fiber bandwidth by providing data channels whose individual bandwidths more closely match those of the electronic devices at their endpoints.

WDM networks can be classified as either *single-hop* or *multi-hop* networks [4,5]. In single-hop (or *all-optical*) networks each message is transmitted from the source to the destination without any optical-to-electronic conversion within the network. Single-hop communication can be realized by using a single wavelength to establish a connection, but such connections may in general be difficult or impossible to find. Alternatively, all-optical wavelength converters may be used to convert from one wavelength to another within the network but such converters are likely to be prohibitively expensive for most applications in the foreseeable future [9].

^{*} This work was supported by the National Science Foundation under grant CCR-9900491 to Harvey Mudd College and grant MIP 96-33729 to the University of Pittsburgh. The authors also gratefully acknowledge the assistance of Mr. Adam Fineman who implemented the algorithms described here and performed the computational experiments.

In *multi-hop* communication networks a message entering an intermediate node on a particular wavelength can be converted into the electronic medium by a receiver and retransmitted on a new wavelength by a transmitter. Each conversion of the message from one wavelength to another is called a *hop*. Multi-hop networks have been shown to enjoy higher utilization of bandwidth and lower probability of blocking than single-hop networks [1]. However, a multi-hop connection may use more transmitters and receivers than a single-hop connection and, depending on the network architecture, each hop can contribute significantly to the communication latency. Therefore, it is generally desirable to find multi-hop connections that minimize the number of transmitters and receivers and/or number of hops used.

Finally, a network may support *unicast* (or *one-to-one*) communication as well as *multicast* (or *one-to-many*) communication. Multicast communication is used in distributed shared memory clusters to support operations such as cache invalidation and is used in wide-area networks for video distribution and teleconferencing among other applications.

In this paper we consider multicast communication in multi-hop circuit-switched networks. We assume networks with an arbitrary number of nodes, a fixed number of transmitters and receivers at each node, and a fixed number of wavelengths on each link. Multicast communication requests are made and released over time. A multicast connection may be realized by constructing a multicast tree which distributes the message from the source node to all destination nodes such that the wavelengths used on each link and the receivers and transmitters used at each node are not used by existing circuits.

The *routing and wavelength assignment (RWA)* problem is that of selecting a multicast tree, the wavelengths on the links in the tree, and thus the intermediate nodes that will perform wavelength conversion. In the *wavelength assignment (WA)* problem a multicast tree is given and the problem is that of selecting the wavelengths on the links in the tree and the intermediate nodes for wavelength conversion.

In this paper we show that although the RWA problem in this model is, in general, NP-complete, the WA problem can be solved in linear time. In addition, we show that the linear time WA algorithm can be extended to find “optimal” solutions under various definitions of optimality such as minimizing the maximum number of hops.

Various aspects of multicasting in WDM networks have been investigated recently for both packet- and circuit-switched networks [3,6,7,8,10]. In work most closely related to the results described here, Kovačević and Acampora [2] have investigated the WA problem for multi-hop unicast routing in circuit-switched meshes and Sahasrabudde and Mukherjee [6] have formulated the RWA problem for multi-hop multicast routing in packet-switched networks as a mixed-integer linear programming problem.

The remainder of this paper is organized as follows. In Section 2 we formally describe the model under consideration and define notation. In Section 3 we give a linear time algorithm for the wavelength assignment problem and generalize

the algorithm to find “optimal” multicasts. Section 4 describes experimental results using these algorithms. Conclusions are given in Section 10.

2 Model and Notation

We represent an interconnection network by a connected directed graph $G = (V, E)$ where the vertices represent switches and the directed edges represent links between pairs of switches. Each switch may be connected to a node or network access station. Except where the distinction is necessary, we henceforth use the terms “switch”, “node”, and “vertex” interchangeably and let n denote $|V|$. Similarly, we use “link” and “edge” interchangeably. Each link can carry some number, w , of different wavelengths denoted by $\Lambda = \{\lambda_1, \dots, \lambda_w\}$. Each node v has $T(v)$ tunable transmitters and $R(v)$ tunable receivers, each of which can tune to any of the w wavelengths. Let $d_{\text{in}}(v)$ and $d_{\text{out}}(v)$ denote the number of incoming and outgoing links, respectively, at node v . We assume that the number of nodes n in the network is variable but that parameters w , $T(v)$, $R(v)$, $d_{\text{in}}(v)$, and $d_{\text{out}}(v)$ are bounded by constants dictated by the technology.

A wavelength on an input link may be routed to the same wavelength on any number of output links and, optionally, to a receiver at the local node. Similarly, a message transmitted on a particular wavelength by a transmitter at a node may be routed on this wavelength to any number of output links. Routing must satisfy the constraint that two messages using the same wavelength cannot share the same link. A switch model with these properties is shown in Figure 1. Switches with some similar characteristics were described by Kovačević and Acampora [2] and by Sahasrabuddhe and Mukherjee [6]. We note that the results described in this paper can be adapted to a number of other switch models.

A *multicast communication request* is an ordered pair (s, D) where $s \in V$ is the source of the multicast and $D \subseteq V - s$ is the set of destination nodes. We assume that multicast communication requests are made and released dynamically. At the time that a particular multicast communication request is made there may be some limits imposed on the routing resources available in the network. Specifically, each node v has some available number $t(v)$ of transmitters and $r(v)$ of receivers that can be used to implement the multicast where $t(v) \leq T(v)$ and $r(v) \leq R(v)$. In addition each link (v, x) has some set $w(v, x) \subseteq \Lambda$ of available wavelengths. Let $W(v)$ denote the total number of distinct wavelengths available on all outgoing links from node v . These resource limits may reflect the actual available resources, due to utilization of resources by existing connections, or these limits may be imposed in order to reduce cost or to leave resources available for subsequent connection requests.

Due to these resource constraints, it may not be possible to realize a multicast communication request. Moreover, in some cases it may not be possible to realize a request when only one wavelength may carry the message on each link, while the connection may be realizable when multiple wavelengths are permitted to carry the message on the same link. Let ℓ denote the maximum number of wavelengths that may be used to transmit the same message over any single

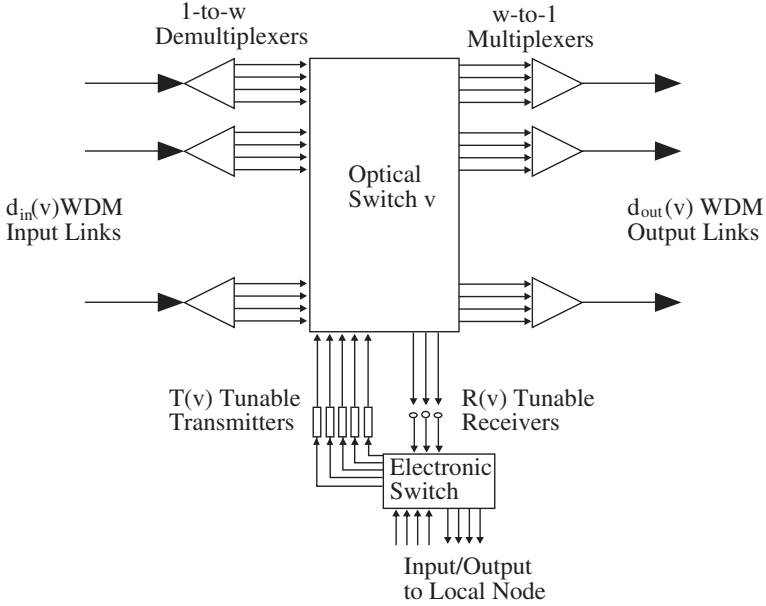


Fig. 1. Schematic of switch model.

link. Figure 2 illustrates an example of a network in which node s is the source of the multicast and all the remaining nodes are destinations. In this example, node s has two transmitters while all remaining nodes have zero transmitters. When $\ell = 1$, node s may use only a single wavelength on each link. Since node u has no transmitters, it is not possible for the message to be delivered to both destinations w and x . On the other hand, when $\ell = 2$ node s may transmit on both wavelengths λ_1 and λ_2 over each link. In this case, all destination nodes can be reached.

We now formalize the definitions of the RWA and WA problems.

Definition 1. Let $G = (V, E)$ be a directed graph and (s, D) a multicast communication request in this graph. A routing and wavelength assignment (RWA) is a collection of links, wavelengths on these links, and wavelength settings for transmitters and receivers at each node such that: each $v \in D$ receives the message from s , at most ℓ wavelengths from $w(v, x)$ are used on each link $(v, x) \in E$, and no more than $t(v)$ transmitters and $r(v)$ receivers are used at each node $v \in V$.

Definition 2. Let $G = (V, E)$ be a directed graph, (s, D) a multicast communication request in this graph, and τ a subtree of G with root s and containing all vertices in D . A wavelength assignment (WA) with respect to τ is a set of wavelengths on the links in τ and wavelength settings for transmitters and receivers at each node in τ such that: each $v \in D$ receives the message from s , at most

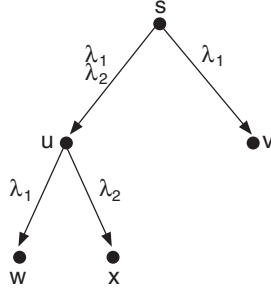


Fig. 2. Node s has two transmitters and all other nodes have no transmitters.

ℓ available wavelengths from $w(v, x)$ are used on each link (v, x) in τ , and no more than $t(v)$ transmitters and $r(v)$ receivers are used at each node v in τ .

The following two theorems show that although the RWA problem can be solved efficiently for some special cases, the RWA problem is, in general, NP-complete. The proofs of these results are omitted in the interest of space.

Theorem 1. *For any value of $\ell \geq 1$, if $t(v) \geq W(v)$ and $r(v) \geq 1$ for all $v \in V$ then a RWA can be found, or it can be determined that none exists, in time $O(n)$.*

Theorem 2. *For any $\ell \geq 1$, if $t(v) < W(v)$ for some nodes $v \in V$ then the problem of determining if there exists a RWA is NP-complete.*

3 The Wavelength Assignment Problem

In this section we show that the wavelength assignment problem can be solved in linear time. Throughout this section, the following assumptions are made:

1. A fixed multicast tree is given with source node s at the root. All destination nodes are in the tree, although the tree may also contain non-destination nodes.
2. All leaves in the multicast tree are destination nodes. (Otherwise leaf nodes can be repeatedly removed until this property is true.)
3. For each destination node v in the multicast tree, $r(v) > 0$. (Otherwise no wavelength assignment exists.)

We begin in Subsection 3.1 by examining the case that $\ell = 1$. In Subsection 3.2 we show how the algorithm can be adapted to find wavelength assignments that minimize the maximum number of hops from the source to all destinations.

3.1 Wavelength Assignment for $\ell = 1$

The algorithm is based on dynamic programming. For each non-root node v , let $p(v)$ denote the parent of v in the given multicast tree. Then $(p(v), v)$ denotes the link from the parent of v to v . Define the predicate $m_v(\lambda) \rightarrow \{\mathbf{true}, \mathbf{false}\}$ by $m_v(\lambda) = \mathbf{true}$ if and only if wavelength λ is available on the link $(p(v), v)$ and node v can deliver the message to all destinations in its subtree if it receives the message on wavelength λ . Recall that every leaf is a destination node. Thus, from the above definition it follows that for each leaf v in the tree,

$$m_v(\lambda) = \begin{cases} \mathbf{true} & \text{if } \lambda \in w(p(v), v) \\ \mathbf{false} & \text{otherwise} \end{cases} \quad (1)$$

In other words, if v is a leaf then $m_v(\lambda)$ is **true** if and only if wavelength λ is available on the link from v 's parent to v .

Next, consider an internal non-root node v which has no receivers available. Since $r(v) = 0$, node v may forward the message on the incoming wavelength to its children but it may not receive the message and then retransmit it on other wavelengths. Let $C(v)$ denote the set of children of v . Let $\bigwedge$ and $\bigvee$ denote the boolean “and” and “or” operators respectively. If $r(v) = 0$,

$$m_v(\lambda) = \begin{cases} \bigwedge_{x \in C(v)} m_x(\lambda) & \text{if } \lambda \in w(p(v), v) \\ \mathbf{false} & \text{otherwise} \end{cases} \quad (2)$$

This rule asserts that v can deliver a message received on wavelength λ to all destinations in its subtree if and only if λ is available on the link entering v from its parent and each child x of v can deliver the message to all destinations in its subtree if x receives the message on wavelength λ .

Next, consider the case that $r(v) > 0$. In this case, node v can use wavelength λ to deliver the message to its children and, in addition, node v can receive the message and retransmit the message to its children using up to $t(v)$ wavelengths other than λ . Define a *wavelength selection set* with respect to λ to be a subset of Λ which contains λ . Let $\mathcal{A}_{\lambda, c}$ denote the set of all wavelength selection sets with respect to λ of size at most $c + 1$. Thus, every set in $\mathcal{A}_{\lambda, c}$ comprises λ and up to c additional wavelengths. Then

$$m_v(\lambda) = \bigvee_{A \in \mathcal{A}_{\lambda, t(v)}} \bigwedge_{x \in C(v)} \bigvee_{\lambda' \in A} m_x(\lambda') \quad (3)$$

if $\lambda \in w(p(v), v)$ and otherwise $m_v(\lambda) = \mathbf{false}$. This rule asserts that $m_v(\lambda)$ is **true** if and only if wavelength λ is available on the link entering v from its parent and there exists some wavelength selection set A comprising λ and up to $t(v)$ additional wavelengths (to be transmitted at v) with the following property: Every child x of v can deliver the message to all of its descendant destinations if it receives the message on one of the wavelengths λ' in set A .

Finally, consider the case of the root node s . Unlike the other nodes in the tree, node s does not receive the message from a parent node. Instead, node s transmits the message using up to $t(s)$ different wavelengths. Let $\mathcal{B}_c$ denote the

set of all subsets of A of size at most c . Define $M = \mathbf{true}$ if and only if a WA exists originating at the source node. Then,

$$M = \bigvee_{B \in \mathcal{B}_t(s)} \bigwedge_{x \in C(s)} \bigvee_{\lambda' \in B} m_x(\lambda') \quad (4)$$

This rule is analogous to the one in Equation (3) except that node s now transmits all wavelengths itself rather than receiving one on an incoming link.

The dynamic programming algorithm is shown in Algorithm 1. Recall that given an acyclic directed graph with n vertices $v_1, \dots, v_n$, a topological ordering of the vertices is a permutation $v_{i_1}, \dots, v_{i_n}$ of the vertices such that if there is a directed edge from v_{i_j} to v_{i_k} then $j < k$. Since the multicast tree is acyclic, there exists a topological ordering of the vertices. Note that by visiting the vertices in the order $v_{i_n}, \dots, v_{i_1}$, a node is only visited if all of its descendants have been visited.

```

Compute a topological ordering  $v_{i_1}, \dots, v_{i_n}$  of the  $n$ 
nodes in the multicast tree
for  $j = n$  down to 1
  Let  $v = v_{i_j}$ 
  for each wavelength  $\lambda$ 
    if  $v$  is a leaf node then compute  $m_v(\lambda)$  using
      Equation (1)
    if  $j > 1$  and  $r(v) = 0$  then compute  $m_v(\lambda)$  using
      Equation (2)
    if  $j > 1$  and  $r(v) > 0$  then compute  $m_v(\lambda)$  using
      Equation (3)
    if  $j = 1$  then compute  $M$  using Equation (4)
  end for (Comment: End inner for loop)
end for (Comment: End outer for loop)
return ( $M$ )

```

Algorithm 1

Note that the actual WA can be found, if one exists, by recording the wavelength assignments in addition to the values of $m_v(\lambda)$ and M .

We now derive an upper-bound on the running time of the algorithm. In general, computing a topological ordering takes time $O(n + m)$ where $n = |V|$ and $m = |E|$. Since our model assumes that the degree of each node is upper-bounded by a constant, $m \in O(n)$ and thus the ordering can be computed in time $O(n)$.

There are a total of wn iterations through the **for** loops. Among the computations performed inside the **for** loops, the computation in Equation (3) requires the largest number of steps. An upper-bound on the number of steps required to compute $m_v(\lambda)$ in Equation (3) can be derived as follows: For each wavelength

λ at most $\sum_{i=0}^{t(v)} \binom{w-1}{i}$ distinct wavelength selection sets are considered because there are $\binom{w-1}{i}$ ways of choosing i wavelengths other than λ from Λ . For each wavelength selection set A , consider the set of children, $C(v)$, of node v . Set $C(v)$ has size at most $d_{\text{out}}(v)$ and for each $x \in C(v)$, at most $t(v) + 1$ steps are required to determine if there exists a wavelength $\lambda' \in A$ such that $m_x(\lambda')$ is **true**. Therefore, in the worst case the number of steps required to compute $m_v(\lambda)$ is bounded by $[\sum_{i=0}^{t(v)} \binom{w-1}{i}]d_{\text{out}}(v)(t(v) + 1)$. Letting $t = \max_{v \in V} t(v) + 1$, $C = \sum_{i=0}^{t-1} \binom{w}{i}$, and $d = \max_{v \in V} d_{\text{out}}(v)$, the running time of the computations performed inside the **for** loops is upper-bounded by $[wCdt]n$. Thus, the algorithm has $O(n)$ running time, with the constant term depending on constants w , t , and d . The impact of these constants on the running time, in practice, is discussed in Section 4.

3.2 Optimal Multicast for $\ell = 1$

In this subsection we show that the dynamic programming solution described in the previous subsection can be adapted to find wavelength assignments which minimize the maximum number of hops required to reach all destination nodes. Similar adaptations can be made for other metrics of optimality.

For each non-root node v , $h_v(\lambda)$ is defined to be the minimum value k such that there exists a path from v to every destination node in the subtree rooted at v which uses at most k hops, assuming the message enters v on wavelength λ . If wavelength λ is not available on link $(p(v), v)$ or it is not possible for v to reach all of the destination nodes in its subtree when the message enters v on wavelength λ then define $h_v(\lambda) = \infty$.

From the definition, it follows that for each leaf v in the tree,

$$h_v(\lambda) = \begin{cases} 0 & \text{if } \lambda \in w(p(v), v) \\ \infty & \text{otherwise} \end{cases} \quad (5)$$

Next, consider an internal non-root node v . If $r(v) = 0$, node v cannot receive and retransmit the message but may only distribute the message to its children using wavelength λ . Thus, if $r(v) = 0$,

$$h_v(\lambda) = \begin{cases} \max_{x \in C(v)} h_x(\lambda) & \text{if } \lambda \in w(p(v), v) \\ \infty & \text{otherwise} \end{cases} \quad (6)$$

If $r(v) > 0$, node v may distribute the message to its children on wavelength λ without incurring an additional hop. In addition, node v may receive the message and retransmit it to its remaining children using up to $t(v)$ wavelengths other than λ . Each child which receives the message on a wavelength λ' other than λ incurs an additional hop. Thus, for $r(v) > 0$,

$$h_v(\lambda) = \min_{A \in \mathcal{A}_{\lambda, t(v)}} \max_{x \in C(v)} \min_{\lambda' \in A} \begin{cases} h_x(\lambda') & \text{if } \lambda' = \lambda \\ 1 + h_x(\lambda') & \text{if } \lambda' \neq \lambda \end{cases} \quad (7)$$

if $\lambda \in w(p(v), v)$ and otherwise $h_v(\lambda) = \infty$.

Let H denote the minimum number of hops required. One hop is incurred by the initial transmission of the message at node s . Therefore,

$$H = 1 + \min_{B \in \mathcal{B}_{t(s)}} \max_{x \in C(s)} \min_{\lambda' \in B} h_x(\lambda') \quad (8)$$

Finally, in Algorithm 1, $m_v(\lambda)$ and M are replaced by $h_v(\lambda)$ and H , respectively, and Equations (1), (2), (3), and (4) are replaced by Equations (5), (6), (7), and (8), respectively. The asymptotic running time and constants are easily verified to be the same as that of the original algorithm.

3.3 Wavelength Assignment for $\ell > 1$

As illustrated in the example in Figure 2, a WA may not exist when each link is permitted to send the message on only one wavelength but may exist when more than one wavelength may be used per link. The algorithms described above can be easily extended to handle the case that $\ell > 1$. The details are omitted in the interest of space.

4 Experimental Results

In this section we describe experimental results using the algorithms presented in the previous section. In the interest of space, we restrict our attention to the case that $\ell = 1$. The first set of experiments used the wavelength assignment algorithm described in Subsection 3.1 to measure the number of multicast requests that were successfully realized as a function of the number of available wavelengths per link and number of available transmitters per node. Specifically, a random multicast tree with 100 nodes ($n = 100$) was generated in which each node had between 0 and 3 children ($0 \leq d_{\text{out}}(v) \leq 3$). The generated tree had height 8 and the destination nodes comprised the 53 leaves of the tree. Each link was assumed to carry 10 distinct wavelengths ($w = 10$). Very similar results to those reported below were obtained for other randomly generated multicast trees with other values of these parameters.

In one group of experiments the number of available transmitters per node was chosen at random from the uniform $[0, 2]$ distribution and in the second group the uniform $[1, 3]$ distribution was used. In all experiments, the number of available receivers per node was set to 1. In each group of experiments the set of available wavelengths on each link was also selected at random where the size of the set was taken from the uniform $[x - 1, x + 1]$ distribution for a given value of x . For each value of x ranging from 2 to 9, 100 runs were performed. The data labeled “Exact Solution” in Figure 3 shows the results of these experiments for the two groups of experiments.

We have noted that the dynamic programming algorithms run in time $O(n)$ but the constant term depends on the number of wavelengths, transmitters per node, and degree of the switches. For the experiments described above, the maximum amount of time required by the dynamic program for a multicast request

was 0.11 seconds on a 450 MHz Pentium 2. However, for larger values of the parameters the running time was significantly larger. For example, for a problem instance with 100 nodes, 32 wavelengths per link, switches of degree 8, up to 3 transmitters available per node for each multicast request, and an average of half of the 32 wavelengths available on each link, the running time increased to 11.38 seconds.

Therefore, in some situations it may be desirable to use heuristics that are faster or simpler than the dynamic programs described here. The exact solutions found by the dynamic programming algorithms can then be used off-line to evaluate the quality of such heuristics. As an example, we have investigated a simple greedy heuristic for finding wavelength assignments. The heuristic operates as follows. The source node, s , determines the available wavelength that can be used to reach the largest number of its children, breaking ties arbitrarily. Then the available wavelength is found that reaches the largest number of remaining children. This process is repeated until a set S of wavelengths is found that can be used to reach all of the children of s . If the number of wavelengths in S exceeds the number of transmitters available at s , the heuristic fails to satisfy the multicast request and terminates. Otherwise each child x of the source node may receive the message on any one of the wavelengths in $S \cap w(s, x)$. For each child x of s , the heuristic determines which $\lambda \in S \cap w(s, x)$ reaches the largest number of children of x . This wavelength is then used to deliver the message from s to x and then from x to as many of its children as possible. Next, the heuristic repeatedly selects the wavelength that can be used to reach the largest number of remaining children of x until all children of x are reachable with the selected wavelengths. If the number of wavelengths selected is larger than the number of transmitters at x then the heuristic fails to satisfy the request and terminates. Otherwise, this process is repeated until all destination nodes are reached.

This heuristic has $O(n)$ running time but a significantly smaller constant term than that of the dynamic program. In comparison to the 11.38 seconds incurred by the dynamic program for the largest problem instance described above, this heuristic required only 0.01 seconds for the same data. The results of running this greedy heuristic for the data used above are shown in Figure 3 for comparison with the exact solutions obtained using the dynamic programming algorithm. Although the exact solutions are generally better than those found by the heuristic, the data also indicates that for some cases the heuristic performs very well. Other more sophisticated heuristics could also be considered at the expense of increased running time.

Next, the dynamic programming formulation from Subsection 3.2 was used to measure the number of hops required for the same parameters used in the above experiments. The results are shown in Figure 4 for the case that the number of transmitters was selected from the uniform $[1, 3]$ distribution. Each curve labeled with a value h indicates the percentage of multicast requests satisfied using at most h hops from the source to any destination. We note that for this data set no multicast request could be satisfied using fewer than 2 hops

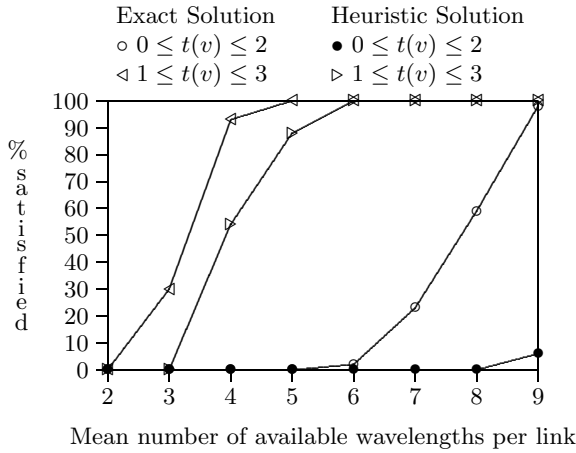


Fig. 3. Percentage of multicast requests satisfied as a function of number of available wavelengths.

and no multicast request required more than 7 hops. These results indicate the relationship between hop counts and percentage of satisfied requests.

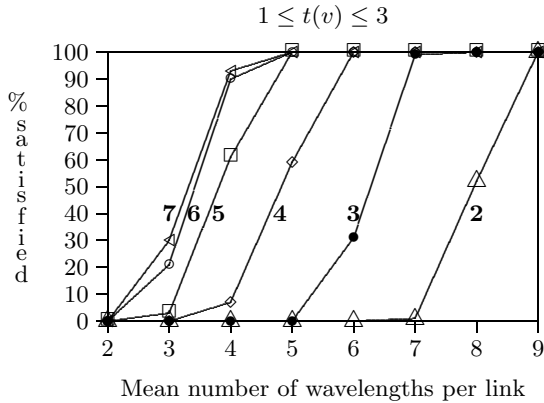


Fig. 4. Percentage of multicast requests satisfied using at most h hops as a function of the number of available wavelengths. Curves are labeled with h .

5 Conclusion and Future Research

In this paper we have investigated the problems of multicast routing and wavelength assignment. We have shown that the wavelength assignment problem for

any fixed multicast tree can be solved in time linear in the number of nodes when the number of wavelengths per link, transmitters and receivers per node, and switch degree are constants. Moreover, we have demonstrated that the dynamic programming algorithm for the wavelength assignment problem can be adapted to find wavelength assignments that minimize the maximum number of hops from the source to all destinations. Similar adaptations can be made to find solutions that are optimal with respect to other metrics.

The algorithms described in this paper can be used either to find exact solutions to the wavelength assignment problem or to evaluate solutions found by faster and simpler heuristics. Heuristics for minimizing the maximum number of hops, transmitter and receiver usage, and other measures of optimality are currently under investigation.

References

1. R. A. Barry and P. Humblet. Models of blocking probability. *IEEE Journal on Selected Areas in Communication*, 14(5):858–867, June 1996.
2. M. Kovačević and A. S. Acampora. Electronic wavelength translation in optical networks. *Journal of Lightwave Technology*, 14(6):1161–1169, June 1996.
3. W. Liang and H. Shen. Multicasting and broadcasting in large WDM networks. In *Proceedings of the 12th International Parallel Processing Symposium (IPPS/SPDP)*, pages 516–523, 1998.
4. B. Mukherjee. WDM-based local lightwave networks, part II: Multi-hop systems. *IEEE Network*, 6(4):20–32, July 1992.
5. B. Mukherjee. WDM-based local lightwave networks, part I: Single-hop systems. *IEEE Network*, 6(3):12–27, May 1992.
6. L. H. Sahasrabudde and B. Mukherjee. Light-trees: Optical multicasting for improved performance in wavelength-routed networks. *IEEE Communications Magazine*, 37(2):67–73, February 1999.
7. S. B. Tridandapani and J. S. Meditch. Supporting multipoint connections in multi-hop WDM optical networks. *Journal of High-Speed Networks*, 4(2):169–188, 1995.
8. S. B. Tridandapani and B. Mukherjee. Channel sharing in multi-hop WDM lightwave networks: Realization and performance of multicast traffic. *IEEE Journal on Selected Areas in Communications*, 15(3):488–500, April 1997.
9. J. M. Yates, M. P. Rumsewicz, and J. P. R. Lacey. Wavelength converters in dynamically reconfigurable WDM networks. *IEEE Communications Surveys*, 1999.
10. Z. Zhang and A. S. Acampora. A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength reuse. *IEEE Journal of Networking*, 3:281–288, 1995.

Improving the Timed Token Protocol

Yacine Bouzida and Rachid Beghdad

Laboratoire "Systèmes Informatiques", UERI
EMP (ex ENITA), BP 17, 16111 Alger, Algérie
Tel: (213) 21 86 34 69
{yacine.bouzida, rbeghdad}@eudoramail.com

Abstract. We present a critical study of "*the timed token*" real-time communication protocol. This protocol presents a drawback towards asynchronous messages. In fact, if all stations of the network have permanent synchronous and asynchronous messages, only the first station can transmit its asynchronous messages during a limited interval of time. Then, only synchronous messages will be transmitted until at least one station does not use all its synchronous capacity for the transmission of synchronous messages. The *regular timed token* protocol [7] has been developed to solve this problem. However, it still occurs in the case where the station uses all its synchronous capacity to send synchronous messages. The proposed here, called *improved timed token*, uses the main key ideas of the two previous ones and permits the transmission of synchronous messages in some critical situations where they cannot be transmitted when using either *timed* or *regular timed token* protocols.

Keywords: local networks, real-time protocol, non-real-time messages, scheduling messages, timed token, regular timed token, scheduling constraints.

1 Introduction

We address the issue of improving the *timed token medium access control (MAC) protocol*. This protocol is suitable for real-time applications not only because of its use in high bandwidth networks but also due to the fact that it has the important property of bounded access time which is necessary for real-time communications. The timed token protocol has been incorporated into many network standards, including the Fiber Distributed Data Interface (FDDI), IEEE 802.4, the High Speed Data Bus and the High Speed Ring Bus (HSDB/HSRB), and the Survivable Adaptable Fiber Optic Embedded Network (SAFENET). Many embedded real-time applications use them as backbone networks.

With the timed token protocol, messages are grouped into two separate classes : the *synchronous* class and the *asynchronous* class. Synchronous messages arrive in the system at regular intervals and may be associated with deadline constraints. The idea behind the timed token protocol is to control the token rotation time. At network initialization time, a protocol parameter called *Target Token Rotation Time (TTRT)* is determined which indicates the expected token rotation time. Each station is assigned a fraction of the TTRT, known as *synchronous capacity*, which is the maximum time for which a station is permitted to transmit its synchronous messages every time it

receives the token. Once a node receives the token, it transmits its synchronous message, if any, for a time no more than its allocated synchronous capacity. It can then transmit its asynchronous messages only if the time elapsed since the previous token departure from the same node is less than the value of T_{TRT} , i.e., only if the token arrived earlier than expected.

The "timed token" protocol presents a drawback towards asynchronous traffic. Indeed, if all the stations of the ring have permanently real-time (synchronous) and non real-time (asynchronous) messages, the first station transmits non real-time messages during T_{TRT} . Thereafter, only the real-time messages will be transmitted until at least one of the stations does not use all its synchronous capacity for the real-time traffic.

The regular timed token protocol was developed to solve this problem. Nevertheless, the problem persists if only one station of the ring has real-time and non real-time messages, it will use all its synchronous capacity to transmit only the real-time messages.

Our contribution : the *improved timed token* protocol brings a solution to the encountered problems.

We present the "timed token" real-time MAC protocol and the regular timed token protocol in section 2. In section 3, we present our approach in details. Section 4 concludes the paper.

2 The Timed Token Protocol

This protocol uses the following parameters :

- T_{TRT} (Target Token Rotation Time) defines the target rotation time of the token.
- $H_k, k=0, \dots, m-1$ (Synchronous capacity of node i), where m is the number of the stations in the ring. This parameter represents the maximum time for which a station is permitted to transmit synchronous messages every time the station receives the token. Note that each station can be assigned a different H_i value. In this paper, we assume that $H_j = H_k, j \neq k \forall j, k \in \{0, \dots, m-1\}$.
- TRT_k (Token Rotation Time). It evaluates the cycle time (this counter is initialized to the T_{TRT} value and re-initialized to this value either when the token arrives early to the station or when the TRT_k is expired).
- LC_k (Late Counter of node k). This counter is used to record the number of times that TRT_k has expired since the last token arrival at node k .
- THT_k (Token Holding Time), defines the time during which the station k may transmit non real-time traffic.

Theoretically, the total available time to transmit synchronous messages, during one complete traversal of the token around the ring, can be as much as T_{TRT} . However, factors such as ring latency Θ and other protocol/network dependant overheads reduce the total available time to transmit synchronous messages. We denote the portion of T_{TRT} unavailable for transmitting synchronous messages by τ .

That is, $\tau = \Theta + \Delta$ where Δ represents the protocol dependant overheads (the token transmission time, asynchronous overrun, etc.). We define the ratio of τ to T_{TRT} to be

α . The usable ring utilization available for synchronous messages would therefore be $(1-\alpha)$ [4].

Thus, a protocol constraint on the allocation of synchronous capacities is that the sum total of the synchronous capacities allocated to all nodes in the ring should not be greater than the available portion of the Target Token Rotation Time (T_{TRT}), i.e.,

$$\sum_{k=1}^m H_k \leq T_{TRT} - \tau \quad (1)$$

In the following studied case (figure 1), we consider $\tau = 0$.

Timed Token Protocol [3]

For each station k , ($k=0, 1, 2, \dots, m-1$) :

$THT_k \leftarrow 0$;

$LC_k \leftarrow 0$; /*initialization procedure */

$TRT_k \leftarrow T_{TRT}$;

Starting the countdown of TRT_k

While the network is working :

If $TRT_k = 0$ then

$TRT_k \leftarrow T_{TRT}$

$LC_k \leftarrow LC_k + 1$

EndIf

At the arrival of the token do :/* data transmission */

0Case

- $LC_k = 0$: /* token early arrival case */

$THT_k \leftarrow TRT_k$

$TRT_k \leftarrow T_{TRT}$

starting the countdown of TRT_k

transmission of real time messages during H_k

starting the countdown of THT_k

While $THT_k > 0$ and ($\exists$ non real-time messages in wait): Transmission of non real-time messages
token passing to the station $(k+1)$ (modulo m)

- $LC_k = 1$: /* token late arrival case */

$LC_k \leftarrow 0$

Transmission of real-time messages during H_k
token passing to the station $(k+1)$ (modulo m)

- $LC_k > 1$: /* error case */
« error recovery » procedure

EndCase

END

Critic :

Let us consider the situation where on one hand all stations have real-time traffic to transmit permanently during their synchronous capacity H_k and on the other hand, the first station uses all the time that it possess to transmit the non real-time traffic (it may transmit during an interval of time corresponding to T_{TRT}) (figure 1).

The next diagram represents the time filling of network transmissions.

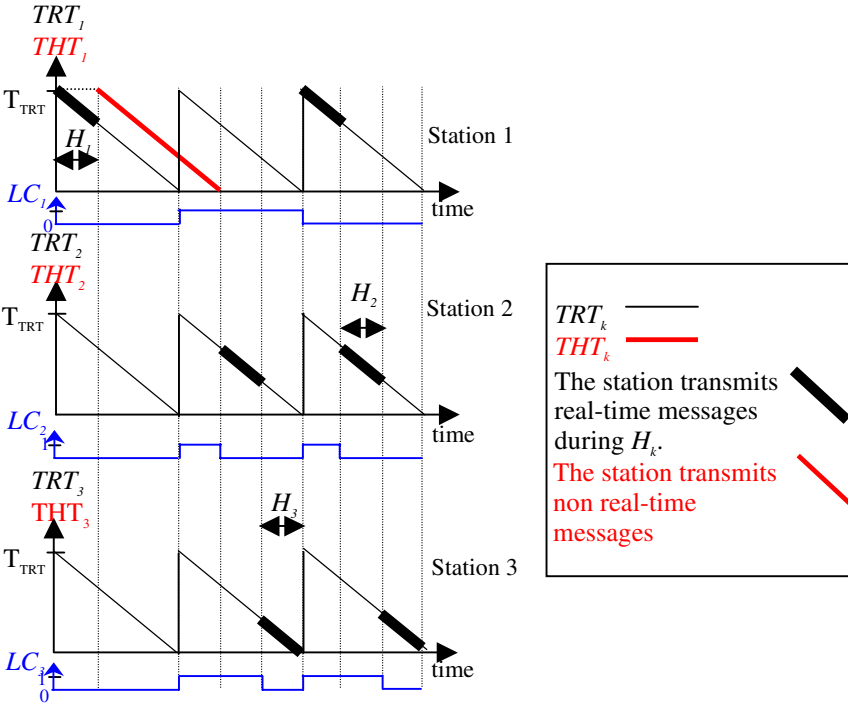


Fig. 1. Example of a critical case with the «timed token» protocol.

- We notice that in the most unfavorable case, all stations use all their synchronous capacity H_k and no longer give a chance to the non real-time traffic. This situation lasts until all stations do not use the totality of their synchronous capacity.
- If all stations respect protocol constraints, no LC_k will be able to reach a value greater than 1 (error situation).

2.1 The Regular Timed Token Protocol [7]

In this algorithm, the author considers T_{TRT} not as the token target time, but as the maximum time.

- We assign to a station k , a synchronous capacity H_k for the real time traffic (if any). And if the synchronous capacity H_k is not expired, the station transfers the non real-time traffic until the expiration of H_k .
- The constraint (1) is valid for this algorithm. For the example of figure 2, we assume that $\tau=0$.

Algorithm of the regular timed token protocol

For each station k , $k=0,1,2,...,m-1$:

$THT_k \leftarrow 0$; $TRT_k \leftarrow T_{TRT}$; /* initialization procedure */
Starting the countdown of TRT_k

For each station k , $k=0,1,2,...,m-1$:

If $TRT_k=0$: «ring recovery procedure» /* error */
At the arrival of the token, Do :/* data transmission */

$THT_k \leftarrow H_k$;

$TRT_k \leftarrow T_{TRT}$;

Starting the countdown of TRT_k and THT_k

while $THT_k > 0$ and (real-time messages in wait) :

transmission of real-time messages

while $THT_k > 0$ and (non real-time message in wait) :

transmission of non real-time messages

passing the token to the station $(k+1)$ (modulo m)

EndDo

End.

Critics :

1 – Let us consider the situation where all the stations of the ring have permanent real-time traffic. Consequently, no asynchronous messages will circulate in the network. However, the timed token algorithm allows the first station to transmit, at the beginning, the non-real-time message during T_{TRT} .

2 – When only one station of the network has, permanently, real-time and non real-time traffic, only real-time messages will be transmitted. This is the main drawback of this variant. (figure 2)

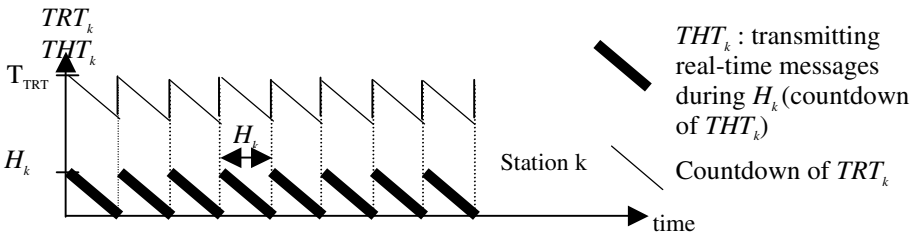


Fig. 2. An example of a critical case with the regular timed token protocol.

The previous diagram shows that no non-real-time messages will be transmitted although all the other stations remain idle.

The proposed protocol uses the same variables as those of the timed token protocol with the introduction of a new variable HR_k that denotes the remaining time from H_k after station k has sent all its real-time messages.

3 The Improved Timed Token Protocol

Principle :

- We assign to each station a time capacity H_k , that represents the maximum time, during which it can transmit the synchronous traffic.
- This protocol allows a station to transmit non real-time traffic whether:
 - it receives the token early, or
 - $HR_k > 0$.

The second principle allows the transfer of asynchronous messages (if any) of the current station, instead of sharing its remaining time HR_k with the other stations or to wait for the next reception of the token to transfer non-real-time messages.

The constraint (1) is still valid for this algorithm. For the example of figure 3, we assume that $\tau=0$.

Algorithm of the improved timed token protocol

```

For each station  $k$ , ( $k=0, 1, 2, \dots, m-1$ )
   $THT_k \leftarrow 0$ 
   $LC_k \leftarrow 0$                                 /* initialization procedure */
   $TRT_k \leftarrow T_{TRT}$ 
  Starting the countdown of  $TRT_k$ 
While the network is working, Do :
  If  $TRT_k = 0$  then
     $TRT_k \leftarrow T_{TRT}$  ;
     $LC_k \leftarrow LC_k + 1$ 
  EndIf
  At the arrival of token Do : /* data transmission */
    Case
      •  $LC_k = 0$                                 /* token early arrival case */
         $THT_k \leftarrow TRT_k$ 
         $HR_k \leftarrow H_k$ 
         $TRT_k \leftarrow T_{TRT}$ 
        Starting the countdown of ( $TRT_k, HR_k$ )
        While  $HR_k > 0$  and (real-time messages in wait) :
          Transmission of real-time messages
         $HR_k \leftarrow THT_k + HR_k$ 
        Starting the countdown of  $HR_k$ 
        While  $HR_k > 0$  and (synchronous messages in wait):
          Transmission of non-real-time messages
        token passing to the station ( $k+1$ ) (modulo  $m$ )

```

```

•  $LC_k = 1$  /* token late arrival case */
   $LC_k \leftarrow 0$ 
   $TRT_k \leftarrow T_{TRT}$ 
   $HR_k \leftarrow H_k$ 
  Starting the countdown of  $(TRT_k, HR_k)$ 
  While  $HR_k > 0$  and (real-time messages in wait) :
    Transmission of real-time messages
  While  $HR_k > 0$  and (non-real-time messages in wait) :
    Transmission of non-real-time messages
  token passing to the station  $(k+1)$  (modulo  $m$ )
•  $LC_k > 1$  /* error */
  « error recovery » procedure
EndCase

```

END

Let us consider the case where we have three stations in the ring such that, at the beginning, the first station uses only a portion of its synchronous capacity to transmit real time messages. After that, all stations possess constrained messages permanently.

The next diagram represents the time filling of the network exchanges for the timed token protocol and the improved timed token protocol :

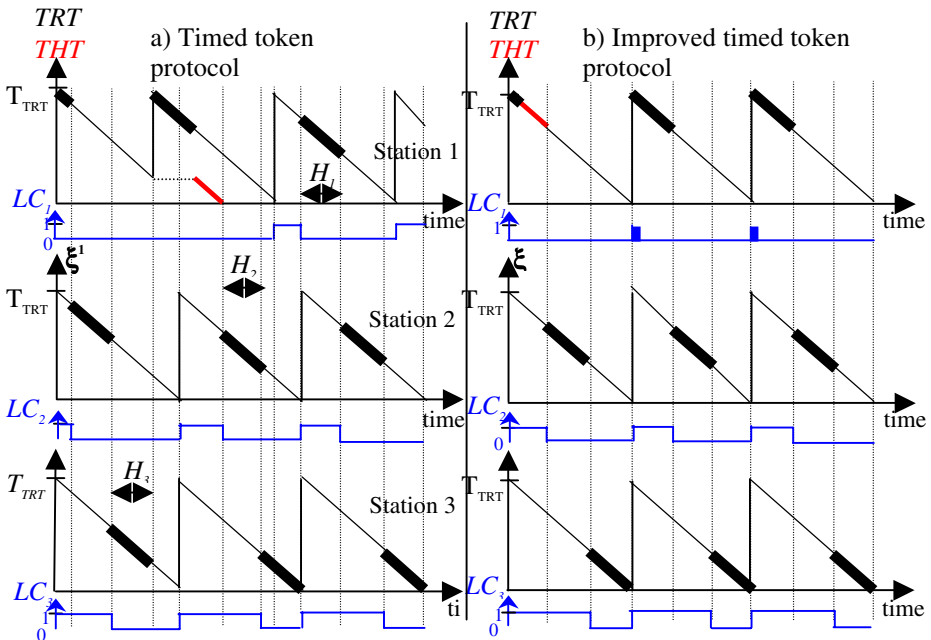


Fig. 3. First comparison between the « timed token » protocol and the improved timed token protocol. a) the first station waits for the next token arrival to transmit non real-time messages. b) In this case, it transmits them immediately. ξ^1 : in this case we assume that LC_i was equal to 1 before the arrival of the token (at the beginning).

According to figure 3, the first station has real-time and non-real-time traffic. In the "timed token" protocol, the non-real-time messages will be transmitted at the second arrival of the token. However, the "improved timed token" protocol will make profit of the remaining synchronous capacity, used for the transmission of the real-time messages, to transmit non real-time messages at the first arrival of the token. Another advantage of this protocol is the transmission of non-real-time messages during the first reception of the token in the case where one station has only non-real-time messages (without waiting for LC_k to be 0). Each station that does not use all its synchronous capacity H_k uses its remaining time HR_k for the transmission of non-real-time traffic before passing the token to the following station (figure 4):

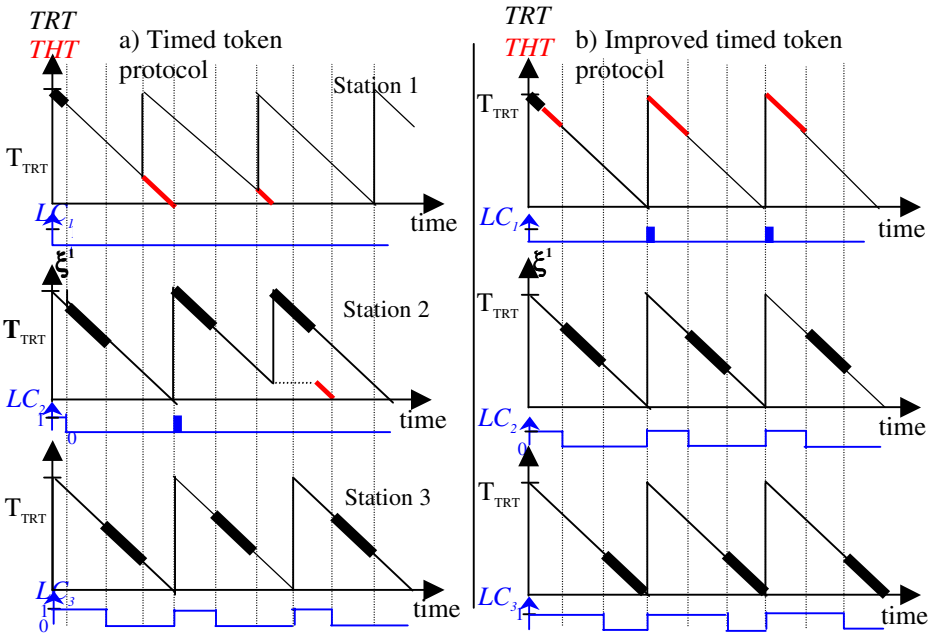


Fig. 4. Second comparison between the « timed token » protocol and the improved timed token protocol. ξ^1 : in this case, we assume that LC_i was equal to 1 before the arrival of the token (at the beginning).

We notice that, under the « the timed token » protocol, station 2 transmits its non real-time messages by exploiting the remaining time of station 1. However, in the improved timed token protocol, the first station transmits its own asynchronous messages (if any) during HR_k before passing the token.

We give now a comparison between the regular timed token protocol and the improved timed token protocol.

Let us consider the situation where there is only one station i that possesses real-time and non-real-time traffic (figure 5):

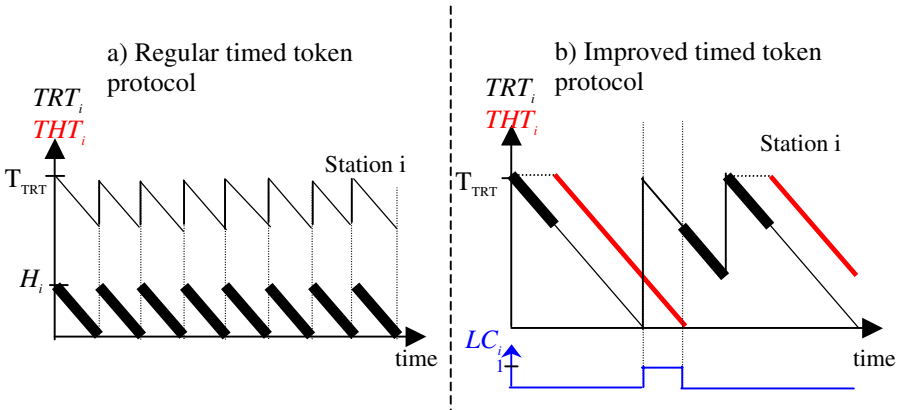


Fig. 5. A comparison between the regular timed token and the improved timed token protocols. a) non real-time messages are not transmitted. b) non real-time messages are transmitted.

Although the station may transmit non time-real messages, it will not be able to do it with the regular timed token protocol, elsewhere it will transmit only real-time messages. With the improved timed token protocol, the transmission of non-real-time messages is guaranteed.

4 Conclusion

We presented a critical study of the timed token protocol. After having given an overview of the timed token protocol and the regular timed token protocol and highlighting their drawbacks towards non real time traffic, we proposed an enhancement to the previous algorithms. The improved timed token protocol allows the transmission of asynchronous messages either without waiting the next arrival of the token or if there is a remaining time after transmitting real time messages.

References

1. G. AGRAWAL, and al. : Guaranteeing Synchronous Messages Deadlines with the Timed Token Medium Access Control Protocol , IEEE transactions on Computers, vol. 43, 3 (1994) 327-339.
2. B. Chen, S. Kamat, and W. Zhao. Fault-tolerant real-time communication in FDDI based networks. In Proceedings of the 16th RTSS, Pisa, Italy, 1995 (141-150).
3. R. GROW : A Timed Token Protocol for Local Area Networks; IEEE Electro 82', Token Access Protocols, 17/3, 1982.
4. J. N. Ulm. A timed token ring local area network and its performance characteristics. Proc.Conf. Local Computer Networks, Feb. 1982 (50-56).

5. M. Hamdaoui and P. Ramanathan : Selection of timed token parameters to guarantee message deadlines. *IEEE/ACM Transactions on Networking*, vol. 3, **3** (1995) 340-351.
6. N. Malcolm and W. Zhao : The timed token protocol for real-time communication. *Computer*, vol 27, **1** (1994):35-41.
7. F. VASQUES de CARVALHO : Sur l'intégration de mécanismes d'ordonnancement et de communication dans la sous-couche MAC de réseaux locaux temps-réel , LAAS Phd Thesis, Toulouse University, June 1996.
8. S. Zhang and A. Burns : An optimal synchronous bandwidth allocation scheme for guaranteeing synchronous messages deadlines with the timed-token MAC protocol. *IEEE/ACM Transactions on Networking*, vol .3, **6** (1995) 729-741.

Design of a Specification Language and Real-Time APIs for Easy Expression of Soft Real-Time Constraints with Java

Keum-Young Sung

School of Computer Science and Electronic Engineering
Handong University, Pohang, 791-940, Korea
kysung@handong.edu

Abstract. With its central and unique ability to execute bytecode on any platform, the Java programming language has gained increasing popularity in a wide area of computation especially in internet-related applications. Even with its broad applicability, the standard Java virtual machine is deficient in the capability to express real-time constraints. In this study, a specification language is suggested to specify real-time constraints, which generates skeletal Java code containing the invocation of real-time APIs. The use of multithreading is proposed to implement real-time APIs without modifying the current Java semantics. This approach may expand the application area of Java applet with existing language features especially to specify soft real-time constraints for visual specification or modeling based on internet. With the suggested technique we can specify timing semantics including maximum, minimum, durational, and relative timing constraints. The detailed execution orders of multithreads to express many forms of timing constraints have been packaged into API libraries for maintainability and readability.

1 Introduction

Specification is concerned with what the software components of a system should do, not with how it is to be implemented [1]. A specification differs from an implementation in that there is no need for it to be efficient in the computational sense; rather a specification describes only the external behavior of the software [2]. A formal specification is easier to understand than a program because it is written in a language chosen for ease of expression [3]. Thus, a formal specification can also be useful for program documentation. To obtain a good specification, a formal specification language is used since it can be tested for ambiguity, consistency, and correctness [4].

The suggested formal specification language temporarily named “Spec” consists of two main parts, i.e., the process list part and the timing constraints part. The necessary

processes used as events are declared in the process part, and the required timing constraints related to the declared processes are specified in the timing part. The processes used in the specification are translated into threads in Java. The syntactic definition of Spec based on the BNF is available in Appendix. The most important role of Spec is to generate appropriate APIs contained in generated skeletal Java classes. The Spec compiler has been implemented with the automatic lexer and parser generator based on the Java programming language, i.e., JLex and CUP [5].

The following example specification shows the real-time requirements for the movement of the robot arm of a welder machine, which has been used as an example in Section 3 of this paper.

```
Spec Machine_Arm
  Begin
    Process
      FirstTimer, SecondTimer,
      FirstEvent, SecondEvent,
      ThirdEvent, FourthEvent, FifthEvent, SixthEvent;
    Timing_Constraints
      MinimumTime:
        FirstEvent, SecondEvent, FirstTimer(5);
      MaximumTime:
        ThirdEvent, FourthEvent, SecondTimer(3);
      DurationTime:
        FifthEvent(4), SixthEvent;
  End Machine_Arm
```

The reserved words of Spec are indicated by italic type in this example. The example specification declares two timers and six events which are supposed to be implemented with multithreading in the target Java classes. In the timing part of the specification, the occurring order of events constrained by a timing constraint is indicated by the actual sequence of events followed by timing information.

The minimum timing constraint stated under the keyword, *MinimumTime*, in the example specification indicates that at least five time units should elapse for the second event, *SecondEvent*, to occur after the occurrence of the first event, *FirstEvent*. The five time units in this case is indicated by the timer process, *FirstTimer(5)*. The role of a timer is explained in detail in Section II. The maximum timing constraint in the example specification indicates that the fourth event, *FourthEvent*, occurs within three time units after the occurrence of the third event, *ThirdEvent*. For the durational timing constraint of the example, the sixth event, *SixthEvent*, occurs after the four time units of the fifth event's duration, *FifthEvent(4)*.

Timing constraints are concerned with the absolute timing of events and the relative order in which actions or stimuli are produced. Absolute timing constraints represent the actual time of the start or finish of an event [6], and are categorized as follows [7]:

- ✂ Maximum timing constraints demand that no more than t amount of time may elapse between the occurrence of one event and the occurrence of another;
- ✂ Minimum timing constraints stipulate that no less than t amount of time may elapse between two events; and
- ✂ Durational timing constraints express that an event may occur for t amount of time.

Real-time systems are those in which the correctness of the system depends not only on the results of computation but also on the time at which the results are produced [8, 9]. Real-time systems are often termed as hard real-time systems, as opposed to soft real-time systems [10]. The degree of dependency on the timing constraints is the only criteria used to differentiate between hard and soft real-time systems. In soft real-time systems, slow response time (or missed deadline) can be tolerated as long as 1) not too many deadlines are missed, and/or 2) deadlines of real-time processes are not missed by much [11].

Currently it is very hard to expect hard real-time performance with Java applet programming mainly due to slow execution speed of the Java virtual machine [12]. To improve the execution speed of Java bytecode, the Just-In-Time (JIT) compiler compiles the bytecode into native code. The result of JIT compilation is cached to be called again when needed [13]. The use of hardware execution of Java bytecode is another approach to run Java code even faster than using JIT compilers [14]. Sophisticated compilers have been tried to conserve system resources and run the cache more efficiently without using dedicated Java chips [15]. The current approach to guarantee the deterministic behavior in multithreading is to use extended Java virtual machine running on a real-time operating system for obtaining improved thread scheduling and run-time performance [16]. Two groups are pursuing to standardize the extensions to Java for real-time embedded applications, which are the J Consortium led by the Hewlett-Packard and the Real Time Experts Group led by Sun Microsystems [17].

2 Absolute and Relative Timing Constraints Expressed with Multithreading

In maximum and minimum timing constraints described in this section, the time span of the timer is assumed to be much longer than the time span of two events bounded by the given timing constraint. In case of relative timing constraint, the durational time span of the second event is assumed to be not shorter than the time span of the first event. The “First_Event” and “Second_Event” used in the timing diagram of all the tables represent the events of the API code snippet, “A” and “B,” respectively.

2.1 Maximum Timing Constraints

To express maximum timing constraints between two events, the multithreaded concurrency is used to express the maximum time span during which the second event

should occur. Table 1 shows the API code snippet and the corresponding timing diagram for implementing absolute maximum timing constraints. The implementation code of the suggested APIs is available when requested to the author. The following shows the specification statement with Spec and the corresponding prototype of the API for a maximum timing constraint:

```
MaximumTime: First_Event, Second_Event, Timer(3);  
  
public static void maximumTimingConstraint(  
    java.lang.Thread First_Event, java.lang.Thread Timer,  
                                java.lang.Thread Second_Event)
```

The join() method called by the timer object after the start of “Second_Event” guarantees that “Second_Event” finishes before the end of the timer duration. The join() method invoked by “First_Event” makes “Second_Event” to get started after “First_Event.” The durations of “First_Event” and “Second_Event” are assumed to be much less than the time span of the timer thread. The timing diagram shows that no more than “Max_Time” amount of time may elapse between the occurrences of two events, “First_Event” and “Second_Event.” The timer event is used to count down “Max_Time” amount of time span.

Table 1. Sequenced order of thread events to express maximum timing constraints

API code snippet	Timing relationship among threaded events
<pre>A.start(); timer.start(); try { A.join(); B.start(); timer.join(); } catch (...) {}</pre>	The diagram illustrates the timing relationship between three threads: First_Event, Second_Event, and a timer thread. A horizontal axis represents time. The duration of First_Event is shown as a rectangle with dots. The duration of Second_Event is shown as a smaller rectangle with dots, starting after First_Event. The timer thread's duration is shown as a dashed rectangle with dots, starting at the beginning of First_Event and ending at the end of Second_Event. A solid double-headed arrow labeled "Max_Time Span" indicates the total time between the start of First_Event and the end of Second_Event. A dashed double-headed arrow labeled "the time span of the timer thread" indicates the duration of the timer thread, which is equal to the Max_Time Span.

2.2 Minimum Timing Constraints

The minimum time that should be passed between two occurrences of events is one of the following three cases:

- 1) Case One: The time span between the start of the first event and the start of the second event;
- 2) Case Two: The time length between the finish of the first event and the start of the second event; and
- 3) Case Three: The time interval between the start of the first event and the finish of the second event.

Table 2. Sequenced thread events to express minimum timing constraints

API code snippet	Timing relationship constrained by minimum timing constraint
<pre>A.start(); timer.start(); try { A.join(); timer.join(); B.start(); } catch (...) {}</pre>	The diagram illustrates a timeline with a horizontal axis labeled 'Time'. Two rectangular events are shown: 'First_Event' (a rectangle with a dotted pattern) and 'Second_Event' (a rectangle with a cross-hatched pattern). A double-headed arrow between the end of 'First_Event' and the start of 'Second_Event' is labeled 'Minimum Time Span of Timer'. Vertical dashed lines extend from the start and end of 'First_Event' to the timeline, and another vertical dashed line extends from the start of 'Second_Event' to the timeline.

For Case One, the durational time of the first event is contained in the minimum timing constraint between two events because the second event is supposed to occur after a minimum time after the start of the first event. For Case Two, a minimum time elapses after the finish of the first event before the start of the second event. The minimum time between two events in Case Three contains both durational times of the two events. Table 2 shows the API code snippet and the corresponding timing diagram for Case Two.

The specification statement and the generated prototype of the API for a minimum timing constraint are as follows:

```
MinimumTime: FirstEvent, SecondEvent, Timer(5);
public static void minimumTimingConstraint(
    java.lang.Thread First_Event, java.lang.Thread Timer,
    java.lang.Thread Second_Event)
```

The program makes use of the join() method called by the timer object to guarantee the finish of the timer object before the start of the second event. The timing diagram shows that the generation of “*Second_Event*” occurs after the elapse of the minimum time period placed by the timer thread.

2.3 Durational Timing Constraints

A durational timing constraint imposes some fixed amount of time on the duration of an event. For example, an event with some duration in a reactive system may activate or deactivate an external device. In this case, the activated or deactivated state of the external device controlled by the signal (event) is constrained by the durational time of the controlling event. Table 3 shows the API code snippet and the corresponding timing diagram.

Table 3. Expression of durational timing constraints

API code snippet	Durational time of two consecutive events
<pre>A.start(); try { A.join(); B.start(); } catch (...) {}</pre>	

As illustrated in Table 3, the `join()` method called by “*First_Event*” guarantees that the finish of “*First_Event*” occurs before the start of “*Second_Event*.” The specification statement and the prototype of corresponding API for a durational timing constraint are as follows:

```
DurationTime: First_Event (4), Second_Event;
public static void durationalTimingConstraint (
    java.lang.Thread First_Event,
    java.lang.Thread Second_Event)
```

2.4 Relative Timing Constraints

There are seven relations between intervals [18] as illustrated in Fig. 1. All the relations in the figure can be used as the types of relative timing constraints between two periodic or aperiodic events.

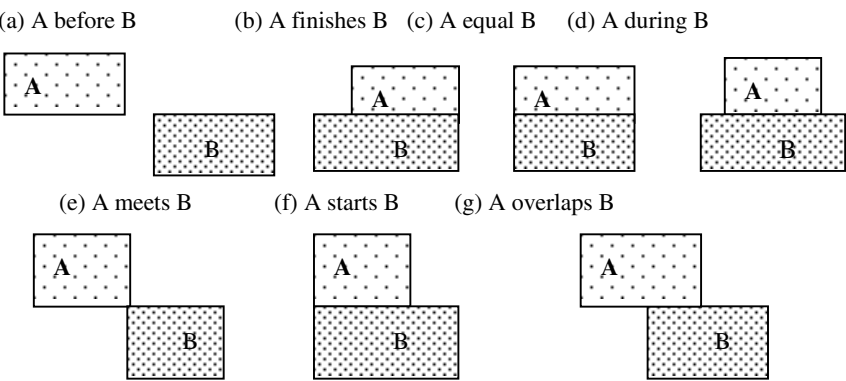


Fig. 1. Seven interval relations between two events, A (*event A*) and B (*event B*)

The relative timing constraints in the form of “*A before B*,” “*A meets B*,” “*A overlaps B*,” and “*A during B*” can be expressed by using the combination of join() method and linearly-sequenced generation of events. However, it is very difficult or almost impossible to express such types of relative timing constraints as “*A equal B*,” “*A finishes B*,” and “*A starts B*” due to the non-deterministic behavior of multithreading and the lack of parallel constructs with Java. Table 4 shows the two types of relative timing constraints, “*A meets B*” and “*A overlaps B*,” which can be applied to express the other types of relative timing constraints, “*A before B*” and “*A during B*.”

Table 4. API code snippets for “*A overlaps B*” and “*A meets B*”

API code snippet for “ <i>A overlaps B</i> ”	API code snippet for “ <i>A meets B</i> ”
A.start(); try { B.start(); A.join(); } catch (...) {}	A.start(); try { A.join(); B.start(); } catch (...) {}

In the API for implementing “*A overlaps B*,” the start of the first event precedes the start of the second event. The finish of the first event occurs before the end of the second event under the assumption that the duration of the second event is not less than the duration of the first event. In the API for “*A meets B*”, the second event is forced to start after the finish of the first event which invokes join() method. When join() of the first event returns, the first event is guaranteed to have finished before the start of the second event. The specification statements and the prototypes of the generated APIs for durational timing constraints, “*A overlaps B*” and “*A meets B*,” are as follows:

```
RelativeTimeMeet: Event_A, Event_B;  
public static void relative_Timing_A_meets_B (  
    java.lang.Thread Event_A, java.lang.Thread Event_B)  
  
RelativeTimeOverlap: Event_A, Event_B;  
public static void relative_Timing_A_overlaps_B (  
    java.lang.Thread Event_A, java.lang.Thread Event_B)
```

3 Example Visual Specification of a Welder Robot Arm

The example machine to be prototyped with Java applet consists of an upper arm, a lower arm, and a welder tip. The moving steps for the example machine are as follows:

- (1) The machine stretches the lower arm after the elapse of the predefined minimum time, five time units, as illustrated in (b) of Figure 2;

- (2) The tip is stretched within the predefined maximum time span, three time units, as illustrated in (c) of Figure 2; and
- (3) The stretched tip is retracted to the lower arm after the predefined durational time span, four time units, as illustrated in (d) of Fig. 2.

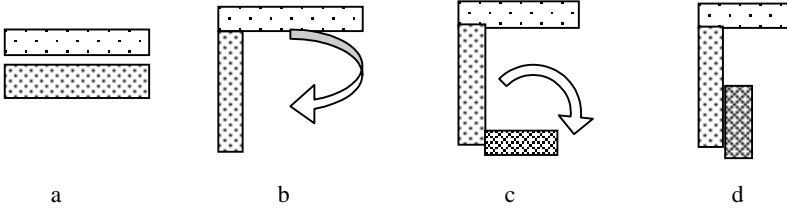


Fig. 2. Prototype of a welder robot arm with the movement sequence of *a* (initial state) , *b* (stretching arm), *c* (stretching tip), and *d* (retracting stretched tip)

The example specification in Section 1 is used to specify the real-time requirements needed for the movement of the example robot arm. The part of the specification related to the real-time requirements of the robot arm is as follows:

Timing_Constraints

MinimumTime:

FirstEvent, SecondEvent, FirstTimer(5) ;

MaximumTime:

ThirdEvent, FourthEvent, SecondTimer(3) ;

DurationTime:

FifthEvent(4) , SixthEvent ;

The above specification statements generate Java skeletal classes containing the three real-time APIs for simulating the robot arm as follows:

- ① `minimumTimingConstraint(First_Event, First_Timer, Second_Event);`
- ② `maximumTimingConstraint(Third_Event, Second_Timer, Fourth_Event);` and
- ③ `durationTimingConstraint(Fifth_Event, Sixth_Event).`

The Java skeletal classes generated by Spec compiler for this example can be obtained from the author when requested. The API calling, *minimumTimingConstraint*, draws the stretched lower arm after the predefined minimum time specified by the timer, *FirstTimer*. The second API, *maximumTimingConstraint*, draws the protrusion of the welder tip within the predefined maximum time specified by the timer, *SecondTimer*. The retraction of the tip after some durational time specified by *Fifth_Event* is specified by calling the API, *durationTimingConstraint*.

4 Conclusions

Using the “Spec” specification language, the functional requirements of a real-time system can be specified without mental burden of raw Java programming. The Spec compiler generates the skeletal Java classes containing real-time APIs. The suggested real-time APIs using multithreading to express soft real-time constraints with Java applet do not affect and complicate the current syntax and semantics of Java. The use of the supposed specification language and the added capability to express timing constraints with Java applet may expand the applicability of the language to web-based visual specification or modeling requiring real-time behavior. The use of the suggested APIs to encapsulate the implementation of timing constraints also keeps modular maintainability of the Java programming language.

To express hard real-time constraints, there is a need to improve the Java virtual machine for overcoming the problems related to the quasi-parallelism, automatic garbage collection, and non-determinism of multithreading. The fact that Java does not support the true parallelism indicates that there is some delicate scheduling complexity with using the concurrency of multithreading to express timing constraints.

Currently proprietary extensions to the Java virtual machine and native methods are used in the application emphasizing the speed of code. Even though the current standard Java virtual machine does not support hard real-time performance, the suggested specification technique and the Java applet with real-time APIs can be used to express soft real-time visual specification of mechanical movement. More research is needed to refine the specification technique and to expand the functionality of APIs to abstract real-time constraints for many thread events which are supposed to prototype the movement of multiple separate machine components related to each other with timing constraints.

References

1. B. Cohen, W. T. Harwood, M. I. Jackson: *The Specification of Complex Systems*, Addison-Wesley Publishing Company (1986)
2. M. S. Feather: Program specification applied to a text formatter, *IEEE Trans. Software Eng.*, Vol SE-8, No. 9, September (1982) 490-498
3. B. H. Liskov, V. Berzins: An appraisal of program specifications, *Research Directions in Software Technology*, P. Wegner (edn.), MIT Press, (1979) 276-301
4. B. W. Boehm: Verifying and validating software requirements and design specification, *IEEE Software*, Vol 1, No. 1, January (1984) 75-88
5. A. W. Appel: *Modern Compiler Implementation in Java*, Cambridge University Press (1998)
6. F. Jahanian, A. K. Mok: Safety analysis of timing properties in real-time systems, *IEEE Trans. Software Eng.*, Vol. SE-12, No. 9, September (1986) 890-904
7. B. Dasarthy: Timing constraints of real-time systems: Constructs for expressing them, methods of validating them, *IEEE Trans. Software Eng.*, Vol. SE-11, No. 1, January (1985) 80-86

8. C. Ghezzi, D. Mandrioli: On eclecticism in specifications: A case study centered around Petri nets, in Proceedings of the 4th International Workshop on Software Specification and Design, IEEE Computer Society Press, April 3-4 (1987) 216-224
9. J. A. Stankovic: Real-time computing, Byte, Vol. 17, No. 8, August (1992) 155-160
10. K. B. Kenny, K.-J. Lin: Building flexible real-time systems using the Flex language, Computer, Vol. 24, No. 5, May (1991) 70-78
11. E. Kligerman, A. D. Stoyenko: Real-time Euclid: A language for reliable real-time systems, IEEE Trans. Software Eng., Vol. SE-12, No. 9, September (1986) 941-949
12. G. Bollella, K. Russell: JSR#000001 Real-time specification for Java, http://java.sun.com/aboutJava/communityprocess/jsr/jsr_001_real_time.htm (2000)
13. C. Horstmann, G. Cornell: Core Java 1.2 Volume 1-Fundamentals, The Sun Microsystems Press (1999)
14. D. Hardin: The Rockwell JEM Microprocessor Family: An Efficient Platform for Real-Time Embedded Java, <http://www.ajile.com/people/hardin/writings/jemplrtia-1998.html> (1998)
15. P. Wagner: Sun Gambles on Java Chips, Byte, Vol. 21, No. 11, November (1996) 79-88
16. D. Hardin: The Real-Time Specification for Java, Dr. Dobbs's Journal, February (2000)
17. S. Ortiz Jr.: The Battle over Real-Time Java, Computer, Vol. 32, No. 6 (1999) 13-15
18. J. F. Allen: Maintaining Knowledge about Temporal Intervals, Communication of the ACM, Vol. 26, No. 11, November (1983)

Appendix: BNF Syntactic Definition of Spec

Rules and shorthand to read the definition are as follows:

1. terminals in italic type;
2. one or more instances by the unary postfix operator +; and
3. a character class by the notation [...].

```

specification ::= Spec identifier Begin body End identifier
body ::= Process ThreadList ; Timing_Constraints TimingList
ThreadList ::= ThreadList , ThreadName | ThreadName
ThreadName ::= identifier | identifier ( digit )
TimingList ::= TimingList TypeOfTiming | TypeOfTiming
TypeOfTiming ::= TimingType : ThreadList ;
TimingType ::= MaximumTime | MinimumTime | DurationTime
               | RelativeTimeMeet | RelativeTimeOverlap
identifier ::= [a-zA-Z_] [a-zA-Z_]*
digit ::= [0-9]+

```


Feedback-Controlled Traffic Shaping for Multimedia Transmissions in a Real-Time Client-Server System

Gabriel-Miro Muntean and Liam Murphy

Department of Computer Science,
University College Dublin,
Belfield, Dublin 4, IRELAND.

GabrielMiro.Muntean@ucd.ie, Liam.Murphy@ucd.ie

Abstract. The number of applications requiring real-time transmission of multimedia streams over the existing network structure is increasing. Large amounts of digital video, audio and text data of continuous nature and with associated timing constraints are thus being exchanged, making network overloads more likely. Our paper presents a feedback-controlled solution for ensuring a continuous transmission and play-out of MPEG streams even in congested network conditions. A double-channel link (TCP and UDP) allows data to be sent from the server to the client, and control information describing the quality of the transmission to be sent between client and server. The control data is collected by a special unit located at the client which regularly inspects the receiver and driver buffers. The control data received from the client is processed by a feedback manager at the server which controls a transmission shaper. The latter can influence the transmission process by modifying parameters such as the transmission buffer size and the transmission frequency. Experimental results show improved behavior of the system in congested network conditions.

1. Introduction

In the early days of the Internet, the transmission of text-only documents, and later text with static pictures, was considered more than enough by most users. Now requests for multimedia data transfers between the growing number of clients and servers are increasing rapidly. All the components (audio, video, data and text) have to be transferred over the current infrastructure which is still using window-based flow control, unsuitable for transmitting continuous streams. New protocols such as Real Time Protocol (RTP) [1], Real Time Streaming Protocol (RTSP) [2], and Xpress Transport Protocol (XTP) [3] have been deployed in order to try to improve multimedia transmission and streaming. Their main concern is to try to meet the users' Quality Of Service (QoS) requirements. Unfortunately all of them rely on the existing IP network elements which treat all transmitted packets equally in so-called "best-effort" service. Extensive research has been done and different proposals for ensuring QoS have been made, such as Integrated Services, Differentiated Services, Multiprotocol Label Switching, or Constraint Based Routing [4]. The cost of

implementing these mechanisms has resulted in limited use to date by the large majority of users. Thus proposals for trying to ensure a better delivery of continuous media streams over the existing infrastructure are of great interest.

Apart from the continuous nature of the multimedia streams, their large size causes problems. Different compression techniques were developed in order to reduce the quantity of data being transmitted. Among them, Motion Picture Experts Group (MPEG) encoding [5, 6] proved to be one of the best, not only from the compression ratio point of view, but also because its structure allows stream playing even in the case of random packet loss [7]. Unfortunately MPEG transmission is very bursty, and even when compressed the stream size is not small. Its transfer over the network may create congestion, especially when the network is already loaded.

This paper proposes a traffic shaping scheme to improve MPEG stream transmission from a server to a client over an IP network. The mechanism tries to reduce the burstiness of MPEG transmission and to lower the effect of network congestion. It uses feedback control information sent from the client to the server, informing it about the quality of the received stream. The server uses this control data in order to adapt its transmission policy, whilst the client continues to send feedback information. The proposed mechanism has been implemented and tested by a client-server system as described in the next section.

2. The Client-Server System's Overview

The system consists of two applications: the server and the client (Fig. 1).

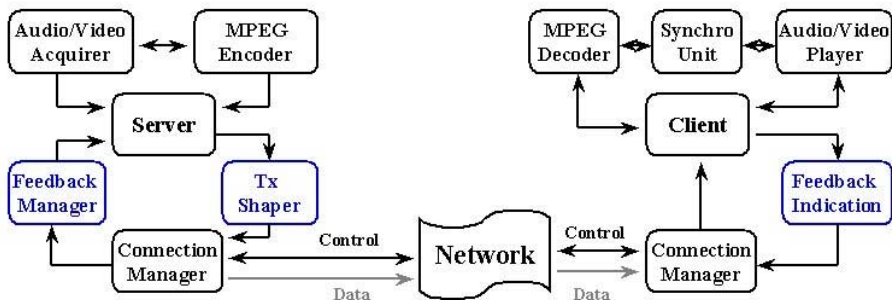


Fig. 1. The structure of the client-server system

The Acquiring Unit and the MPEG Encoder card are used by the server to capture and compress the multimedia streams. The server's Connection Manager helps it to listen for incoming connection requests from the clients, and, once they have been accepted, to maintain them. The latter is also the purpose of the client's Connection Manager. The MPEG Decoder, the Synchronization and the Display and Play Units are in charge of transforming the received data into multimedia streams and of displaying or playing them to the client. The Feedback Indication Unit and the Feedback Manager have an important role in implementing the feedback control

scheme. The communication between the server and client is done via a double TCP and UDP channel described in detail in [8]. This combines the advantages of both protocols: reliability for the control TCP connection, and speed and multicasting for the unreliable UDP channel used for data transfer.

The client-server system has been implemented in Visual C++ 6.0, using an object-oriented approach. It makes use of the multi-threading support offered by the implementation environment. The message and event handling by both the client and the server applications is supported by the Windows event and messaging system. The sockets and all the mechanisms necessary for inter-application communication are provided by the Windows Sockets 2 (WinSock2) architecture. MPEG streaming, buffering, decoding and playing [9], as well as the mechanisms described in detail in this paper, have been implemented and tested by the authors.

3. The Server Transmission Shaper

Processing MPEG streams (decoding, playing, transmission) is very demanding in resources such as CPU, memory and bandwidth [10]. The recent availability of very high-performance CPUs (currently 1.5 GHz) and the continuously-reducing price per megabyte of RAM memory has not been accompanied by a similar significant increase in bandwidth, which still remains the bottleneck for any transmission of multimedia data. The relative sizes of I, P and B MPEG video frame types make the flow very bursty and sensitive to loss and jitter. Besides, in the majority of cases, a big difference between the peak and mean bit-rates adds an extra burstiness to the MPEG video flow [11].

The traffic shaper we propose reduces the burstiness of the transmission by introducing a double control of the sending process. Both buffering mechanism and transmission scheme are supervised and controlled by the Feedback Manager (Fig. 2).

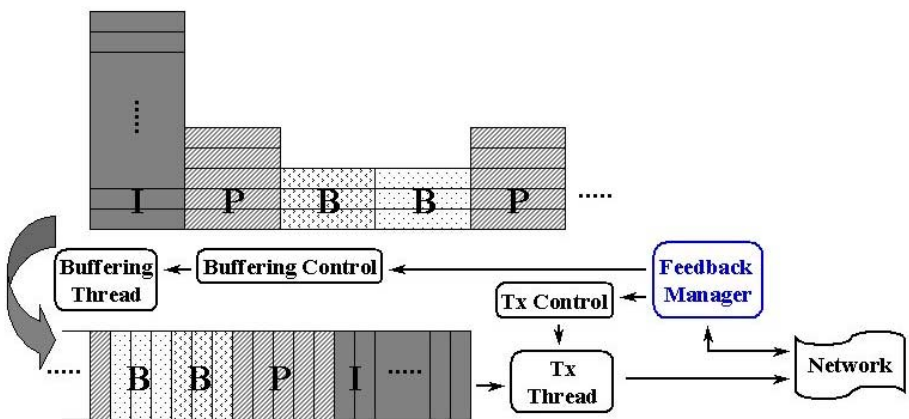


Fig. 2. The Traffic Shaper uses feedback information to adjust the transmission parameters

The scheme implements a modified token bucket mechanism [12] to which a variable token generation procedure has been added. A feedback-controlled timer, part of the Transmission Control Unit, generates tokens used for sending data packets. Once the timer has been started, it continuously generates tokens with a certain frequency. Thus for periods of time the transmission bit-rate can be considered constant. The main advantage of such a scheme is that it reduces the burst caused by sending MPEG video frames, by spreading data to be sent over a larger period of time (Fig. 3). This may add delay to the last packets belonging to some of the frames, but it has been found experimentally that the mean value for the packet delay doesn't differ too much from the one experienced with the normal transmission scheme (Section 5). The client buffering and other features of our feedback scheme compensate for this small disadvantage, and as a matter of fact the computed jitter decreases.

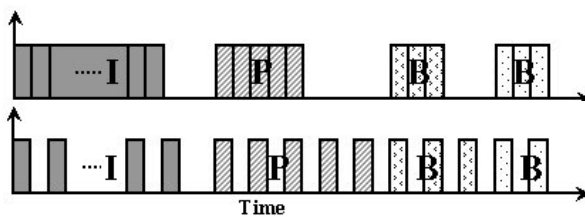


Fig. 3. The transmission traffic shaper reduces the burstiness of the transmitted MPEG stream
a) The normal transmission case b) The transmission with the traffic shaper scheme

Besides its role in reducing the burstiness of the MPEG transmission, the feedback-controlled generation of tokens allows the modification of the transmission bit-rate by varying the transmission control's timer frequency. The timer will be reinitialised, and will generate tokens with a different frequency, any time the Feedback Manager instructs it to, after analysing the feedback data received from the client. The measures it takes depend not only on the current feedback data, but also on their fluctuation in time, thus preventing changing the server's state too frequently. For the case of a multicasting transmission, a simple arbitration scheme has been designed, taking into account which measure is best for the majority of the clients.

The Feedback Manager controls the transmission buffering as well. It is experimentally proved that the effect of losing a packet is more important for the transmitted stream if its size is larger than if a small packet is lost (Fig. 5). Sending large packets also introduces an extra burstiness to the transmission. Smaller packet sizes lead to increased overhead compared to larger packet sizes. Sending packets that are too small not only requires the client to be able to receive them in time to avoid losing them, but also it needs some time to reorder them. This is necessary because it is more likely they will arrive out of order than in the case of larger packets. Thus the Feedback Manager has to dynamically find a trade-off between the transmission frequency and the size of the packets, to maximise the performance of the transmission. For taking its decisions, the Feedback Manager relies on control feedback data received from the Feedback Indication Unit situated at the client and described in the next section.

4. The Client Feedback Indication Unit

The client application consists of a receiver thread (with a higher priority) called by the Windows framework every time when an incoming data packet has arrived, an MPEG decoder thread which decodes received data, and a player thread in charge of displaying and playing decoded data. Two buffers are shared by the client threads: a receiver buffer and a driver buffer for both audio and video streams respectively (Fig. 4).

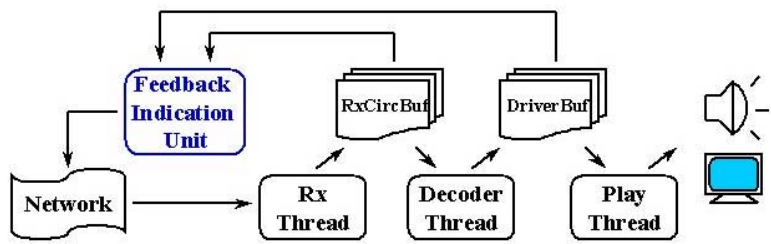


Fig. 4. The client's structure includes a Feedback Indication Unit

The purpose of the Feedback Indication Unit is to collect data about the quality of the transmission where it can be best judged: at the client. It analyses both the receiver buffer and the driver buffer occupancies, and statistical data is updated in real-time about the number of lost frames, frames being late or which come out of order. The Unit repeatedly sends control messages to the server carrying reports about the state of the reception. They are processed by the server's Feedback Manager which takes decisions in real-time which improve the quality of the transmission.

5. Experimental Results

In Fig. 5a we plot the number of lost packets when different transmission packet sizes have been sent. Although the number of lost packets doesn't vary much, it slightly decreases with the size of the transmitted packet. But, as mentioned in Section 3, the percentage of data lost (Fig. 5b) is much higher if large packets don't arrive.

For computing the one-way delay of the packets, we need to have both destination and sender computers with perfectly synchronized clocks. For a very precise synchronization, special devices as GPS, atomic clock or an ISDN synchronous clock board are needed [13]. Our experiments deal with millisecond order delays, so we can use NTP protocol [14] for synchronizing both the server's and the client's clocks, by connecting to the Atomic Clock time server in Boulder, Colorado (USA) and adjusting both computer's clocks to match the atomic clock value.

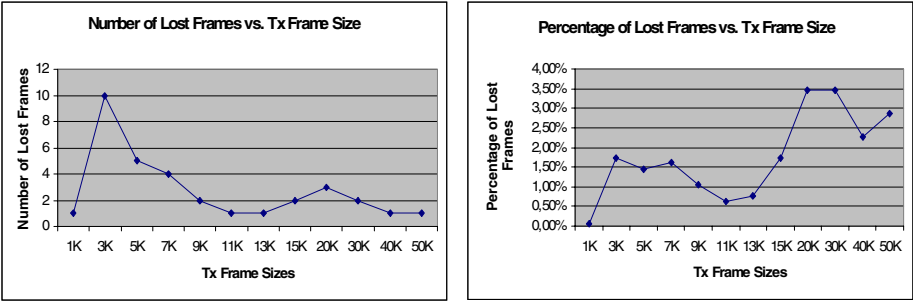


Fig. 5. a) Number of Lost Frames vs. Frame Size b) Percentage of Lost Frames vs. Frame Size

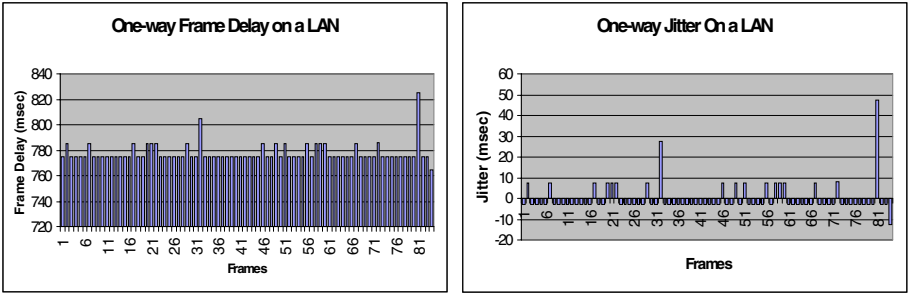


Fig. 6. Transmission Without Traffic Shaping Scheme a) One-way Frame Delays On A Normal LAN b) One-way Transmission Jitter On A Normal LAN

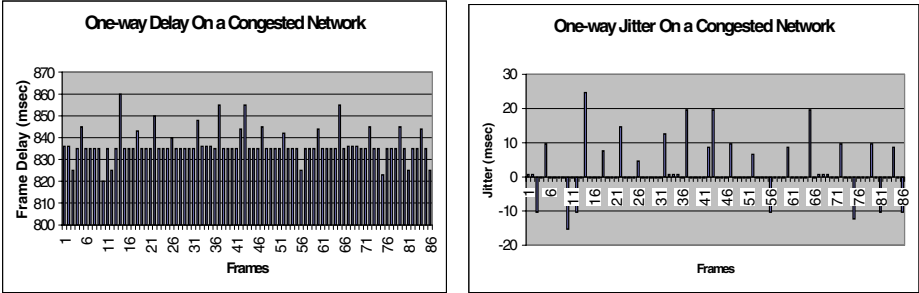


Fig. 7. Transmission Without Traffic Shaping Scheme a) One-way Frame Delays On A Congested LAN b) One-way Jitter On A Congested LAN

To compare the behavior of our a client-server system with and without traffic shaping scheme, we analysed first the results obtained during the transmission of a 9-second MPEG system stream (1.6 Mbytes) over a LAN in two different conditions. Fig. 6 shows the one-way delay and jitter measured in the case of a normally-loaded LAN without traffic shaping while Fig. 7 shows the results obtained in the case of transmission over a congested LAN without traffic shaping.

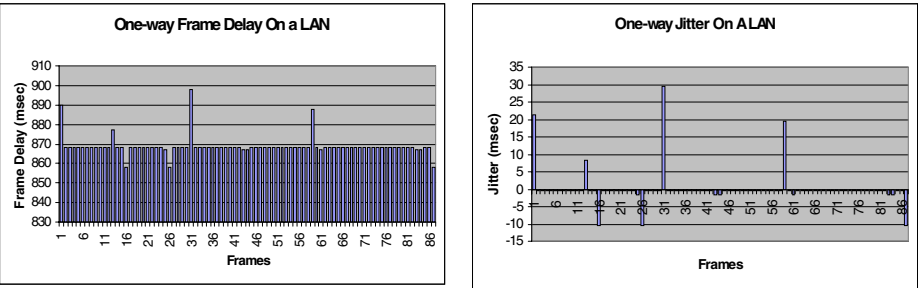


Fig. 8. Transmission With Traffic Shaping Scheme a) One-way Frame Delays On A Normal LAN b) One-way Transmission Jitter On A Normal LAN

In Fig. 8a and Fig. 9a we plot the one-way delays for the same 9-second MPEG system stream over the same LAN in the same two cases (a normally-loaded network and a congested one), this time using the traffic shaping scheme. In both cases, the extra delay added by our traffic shaper is small, and in both cases, the jitter decreases (Fig. 8b and Fig. 9b).

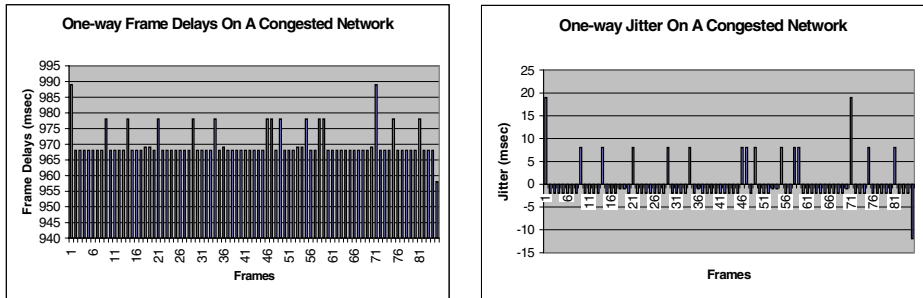


Fig. 9. Transmission With Traffic Shaping Scheme a) One-way Frame Delays On A Congested LAN b) One-way Jitter On A Congested LAN

Our experiments consisted of transmissions of the same MPEG stream over a LAN during the off-peak and peak hours of a day, when the network is normally-loaded and congested, respectively. The client's Feedback Indication Unit took into account both the receiving buffer occupancy and the number of packets lost during the transmission, those which arrived too late for play-out, or out of order. In Fig. 10 we captured the dynamics of the server state during the transmission. The server asymmetrically changes its state after receiving and analysing control messages from the client. Thus if a report of a decreasing reception quality at the client side is received, the server immediately changes its state into a new one with lower quality transmission. In case of a feedback control information carrying news about an improvement in the receiving quality, the server waits for a number of successive positive reports before increasing its state into a higher quality transmission one.

Because the server assumes at the beginning that the highest quality transmission state can be maintained, for all the transmissions, regardless of the state of the

network, we noticed a transition period when the server is searching for its right state. Eventually, after a few transitions between states, the server stabilises and continues the transmission at a certain level of quality. In the case of a congested network (Fig. 10b) this level is lower than the one experienced in a normally-loaded network (Fig. 10a). Also it is more likely to have further state transitions in a congested network than in one with normal loading.

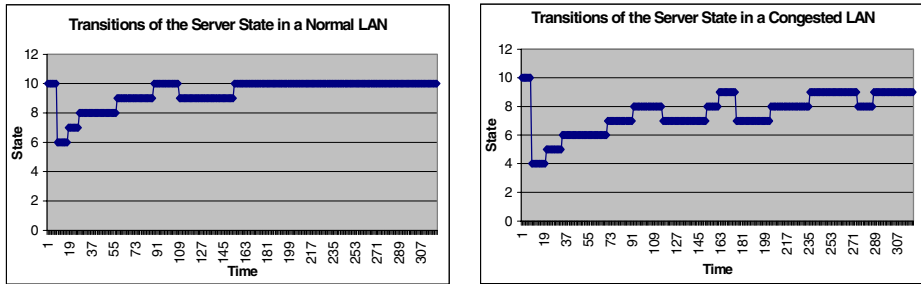


Fig. 10. Server State Transitions a) On a Normal LAN b) On A Congested LAN

6. Conclusions and Further Work

The main idea of this paper is to present a mechanism for traffic shaping of multimedia transmissions driven by the feedback control messages sent by the client. Different criteria were used for generating the reports carried by the feedback control messages. The dynamics of both the receiver and the driver buffer occupancies were taken into account, as well as the number of lost or late packets. The use of other metrics may be the subject of future investigation. The feedback data are collected and analysed by the server which takes decisions in order to improve the transmission. Varying the transmitted packet size and adjusting the transmission bit-rate were taken into account as possible server measures. They are applied effectively by a specially-designed transmission traffic shaper.

The application of our traffic-shaping scheme makes possible a trade-off between the continuity of the transmitted stream and its quality. In the majority of cases it is preferable to continue the streaming and display or play with a worse quality (eventually altered by some lost packets) than to stop the whole stream while buffering.

Some experimental results concerning the transmission packet size and the transmission one-way delay and jitter were analysed. They prove that the proposed traffic shaping scheme is well behaved even in the case of congested network conditions. The testing was done while transmitting over a LAN, but experiments with transmission over WAN are in progress. The results may be improved if some other feedback-controlled measures (such as real-time modification of the MPEG encoding rate) were used in conjunction with the current ones.

References

1. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RFC1889: "RTP: A Transport Protocol for Real-Time Applications", January, 1996: <http://www.ietf.org/rfc/rfc1889.txt>
2. H. Schulzrinne, A. Rao, R. Lanphier: "Real Time Streaming Protocol (RTSP)", April, 1998, <http://www.ietf.org/rfc/rfc2326.txt>
3. Robert M. Sanders, Alfred C. Weaver, "The Xpress Transfer Protocol (XTP) - A Tutorial" ACM Computer Communication Review, Vol. 20, No. 5, October 1990, pp. 67-80.
4. Xipeng Xiao & Lionel M. Ni, "Internet QoS: A Big Picture", IEEE Network, Vol. 13, No. 2, March/April 1999.
5. ISO/IEC International Standard 11172, "MPEG-1 - Coding of Moving Pictures and Associated Audio for Digital Storage Media up to 1.5 Mbits/s", November 1993.
6. ISO/IEC International Standard 13818, "MPEG-2 - Generic Coding of Moving Pictures and Associated Audio Information", November 1994.
7. Didier LeGall, "MPEG: A Video Compression standard for Multimedia Applications", Communications of the ACM, Vol. 34, No. 4, April 1991, pp. 46-58.
8. Gabriel-Miro Muntean, Liam Murphy, "A Novel Feedback Controlled Multimedia Transmission Scheme", International Conference on Telecommunication ICT'01, June 4-7, 2001, Bucharest, Romania.
9. Gabriel-Miro Muntean, Liam Murphy, "An Object-Oriented Prototype System for Feedback Controlled Multimedia Networking", ISSC'2000, Dublin, Ireland, June 29-30, 2000.
10. Ali Marefat, "Use Of Multilevel Traffic Contracts for Admission Optimization of Multimedia Connections in Enterprise Networks", ENM'97, Enterprise Networking Mini-conference, Montréal, Canada, June 8-12, 1997.
11. Steven Gringeri, Khaled Shuaib, Roman Egorov, Arianne Lewis, Bhumi Khasnabish, Bert Basch, "Traffic Shaping, Bandwidth Allocation, and Quality Assessment for MPEG Video Distribution over Broadband Networks", IEEE Network, Vol. 12, No. 6, November/December 1998, pp. 94-107.
12. M. F. Alam, M. Atiquzzaman, M. A. Karim, "Traffic Shaping for MPEG Video Transmission Over The Next Generation Internet", Computer Communications Journal, No. 23, 2000, pp. 1336-1348.
13. T. Yamashita, S. Ono "Synchronizing Clock Frequency with a high speed Digital Network", IEICE Technical Report, CPSY94-119, March 1995.
14. David L. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis", March, 1992, <http://www.ietf.org/rfc/rfc1305.txt>

Bandwidth Reallocation Techniques for Admitting High Priority Real-Time Calls in ATM Networks

Lawrence K. Miller¹ and Ernst L. Leiss²

¹ Department of Electrical Engineering and Computer Science, University of Toledo,
Toledo, OH 43606

lmiller@eecs.utoledo.edu

² Department of Computer Science, University of Houston, Houston, TX 77204
coscel@cs.uh.edu

Abstract. A high priority real-time connection is denied admission to an ATM network if sufficient bandwidth is not available along all suitable paths through the network. Bandwidth reallocation and dynamic active channel re-routing are techniques that can be used to admit high priority real-time connections where traditional CAC techniques would deny admission. A node can select lower priority channels, reallocate their bandwidth to the new higher priority connection being admitted, and reroute those channels so that their QoS requirements and transmission deadlines can still be satisfied. At call admission time, one or more backup channels are established for those primary channels that are likely to be selected as victims for bandwidth reallocation. This allows reroutes to be handled quickly and efficiently. When reroutes occur, the protocols ensure that the transmitted data are received on time and in sequence, which is essential for real-time communications. SANRoP, a cell based discrete event simulator, was developed to simulate these protocols in an ATM network in order to determine how well they perform.

1 Introduction

We examine the problem of call admission of prioritized real-time communications channels with call establishment deadlines in an ATM network. Channel re-routing protocols and the technique of bandwidth reallocation are tools for admitting high priority real-time connections when traditional call admission control (CAC) algorithms would deny admission due to insufficient resources. Since we do not want to renege on prior commitments, two main problems must be solved. First, how can the network admit high priority calls when there is insufficient bandwidth available? Reallocating bandwidth from the lower priority applications can resolve this problem. However, this brings up the second problem. Channels whose bandwidth has been reallocated elsewhere will likely need to be rerouted so that they can also meet their own QoS requirements and their transmission deadlines. However, this is not a trivial problem.

When a call is admitted to a typical ATM network, a path/route is selected through the network, and bandwidth/resources sufficient to meet the Quality of Service (QoS) requirements and the traffic contract are allocated/reserved along that path for the duration of the call. Once allocated to a channel, other channels can't use its

bandwidth until it is released by closing the connection. This means that other calls may not be admissible due to a lack of available resources. Thus, when the network load is near capacity, such as with peak usage times associated with video on demand services, interactive network games, multi-party teleconferencing, etc., then most of the network's resources will likely be tied up and unavailable for high priority real-time connection requests.

In recent work, the issue of channel rerouting has been examined for the purpose of ensuring that the network is fault tolerant in the face of system component failures. A *reactive* method [3], several *forward recovery* methods [4, 5, 12, 19], *static routing* methods using *local detours* [2, 6, 21, 22], and *end-to-end detouring* methods [1, 10, 11, 13, 17], have been studied. These methods are designed for fault recovery and are unsuitable when QoS guarantees and timing constraints associated with real-time channels must be continuously satisfied, even during channel rerouting [15, 16]. However, these techniques can be adapted, with proper route selection for the primary and backup channels to support dynamic channel rerouting.

In this paper, an *end-to-end* detouring approach and two *local* detouring approaches (*node-level* detours and *port-level* detours) are described as methods for rerouting channels. The detours in our methods are chosen to have as minimal an impact as possible on the path lengths/costs of the connections after reestablishment, thus eliminating the problems with meeting the connection's QoS requirements and transmission deadlines (if any). In [15], some initial results of a simulation study were presented which show that node-level detours can be successfully used to admit high priority calls in a network experiencing relatively moderate load conditions. In [16], details regarding the algorithms used for establishing and activating node-level detours were presented in more detail. For this paper, a new simulation study was performed to see how these new protocols performed under moderate loads, when both node-level and port-level detouring were employed.

The rest of this paper is organized as follows. Section 2 presents the technique of bandwidth reallocation for releasing bandwidth for use by higher priority channels. Section 3 presents the algorithms for rerouting channels selected to have their bandwidth reallocated. Section 4 presents methods for route and detour selection to support these techniques while minimizing resource allocation / consumption. Section 5 presents the results of the simulation study on the above three mentioned detouring approaches. Section 6 concludes with a discussion of future work and a summary of the work presented in this paper.

2 Bandwidth Reallocation

When a high priority real-time channel is requested and insufficient bandwidth exists on all paths through the network capable of supporting the channel, then the network must either obtain the necessary bandwidth for the channel or deny the request for access to the network. Since denying the request is not a desirable result for high priority connections, some means of obtaining bandwidth must be employed. The naïve solutions for admitting high priority real-time channels by statically pre-reserving a percentage of each link's capacity, or by renegotiating the QoS guarantees associated with low priority channels, are often not feasible.

The solution adopted here is to use *bandwidth reallocation* [15, 16]. Bandwidth can be stolen from existing connections starting with the lowest priority connections first. Existing lower priority connections are selected and their bandwidth is reassigned to the higher priority channel by the call admission control algorithm until enough bandwidth is freed to establish the high priority connection.

Sufficient time must be allowed for: 1) the network to activate (and maybe even establish) the low priority channel's backup channel for this node's local detour, and 2) the deprecated sub-channel just bypassed to be drained of its residual cell streams, before the bandwidth can be transferred. The time necessary to accomplish these two tasks will cause the CAC algorithm, trying to admit the high priority channel, to incur additional delay. The techniques described in this paper, and in [15, 16], attempt to minimize this delay so that call setup deadlines can still be met.

3 Channel Rerouting

Rerouting a channel depends upon whether the connection is still active after its bandwidth has been reallocated. Inactive channels do not need to be rerouted immediately, but may be rerouted eventually after sufficient time has elapsed. Active channels, however, must be immediately rerouted so that their QoS guarantees and transmission deadlines can still be met.

Rerouting an active channel is more problematic than that of an inactive channel because of the delay requirements associated with them. Most likely there is insufficient time to allow the channel to drain before rerouting it. The reallocation point sends signals notifying the detour points of the channel reroute. Once the detour is activated, the deprecated part of the channel must be allowed to drain its stream of cells before being torn down. Unfortunately, this introduces a problem. It is now the case that two parts of the same channel are traversing partially independent paths through the network, and could end up being delivered out of order at the destination.

Our method uses *delimiters* to allow the detour points to distinguish how the cells arrived, i.e. whether they traversed the original route, or the detoured route. The detour points buffer cells arriving from the detoured route until the end of the cell stream arrives from the deprecated subpath, i.e., the *channel end-stream identifier cell* is seen. With proper path layout, the size of the buffer allocated to this channel will not be very large, thus having a negligible impact upon network resources.

4 Channel Admission

The selection of the routes for the primary path and all of its detours depends upon the priority of the channel being established, and whether the end-to-end transmission deadlines of the primary subpath, can be satisfied by some detour. Low priority channels can tolerate increases in path length/cost which have a negligible impact on how the network continues to meet the QoS guarantees. Thus, its primary path can be a shortest "cost" path through the network, and the detours can be selected to bypass the primary subpaths accordingly. For really low priority channels, it may be possible to use a single alternate route, i.e. an end-to-end detour.

For high priority real-time channels, a different approach is necessary. These types of channels don't tolerate increases in path length/cost very well, if the network still wants to meet their QoS guarantees and transmission deadlines (very small increases may be tolerated with changes in hop by hop deadlines along the local detour, which is an exception handled by the algorithm). Thus, if these channels are potential victims, then a sub-optimal path through the network should be chosen so that the shortest "cost" path can be used as an end-to-end detour in order to maintain, or even reduce, the path length/cost when it replaces the respective primary path.

4.1 Local Detour Path Selection

In addition to selecting a primary path at call admission time, a set of alternative backup path candidates for use as local detours are chosen for use in the event of bandwidth reallocation. An attempt is made to find multiple detours for every primary subpath of the primary path so that the *best* option can be selected for use at reroute time in order to continue to meet the QoS requirements and any transmission deadlines of the connection being rerouted. These local detours are chosen so that they don't use any nodes or links that are involved in the primary path, except for the two detour end-point candidates for which the path detour is being selected.

In [15, 16] algorithm are presented for finding A *Local Detour Set* which is a set of local detours, each of which routes around the same specific node of the primary path. Once a Local Detour Set is computed for each node of the primary path, the "best" candidate is selected from this set as the detour around that node, should it be needed by a reroute operation. For efficiency, each node in the primary path can calculate its own local detour set and select its best detour during the call admission process.

4.2 Primary Path Selection

It is desirable to select a path which does not deviate too much from the cost of the optimal/shortest path, as this can have a negative impact on the network by overly wasting allocated resources, and may even be unsuitable for the channel (i.e. the end-to-end delay may be too great to meet the channel's requirements). The path selected may be an optimal path, or it may be a sub-optimal path.

A simple path selection algorithm [15, 16], is used to find acceptable sub-optimal paths through the network. It finds the shortest *cost* path through the network, and disallows it. The algorithm then finds another *shortest cost path* through the network that detours around the disallowed path. The algorithm is a very simple heuristic that finds a path through the network, which is not the shortest *cost* path. This heuristic algorithm is used because finding an optimal set of paths through the network subject to a set of multiple constraints is known to be a NP-complete problem. If more extensive parameters need to be considered in choosing a primary path, then an algorithms such as the ones in [7, 14, 18, 20], which take into consideration throughput, delay, and error rate, can be used to find routes through the network. Although the shortest *cost* path is explicitly disallowed as the primary path, it may be one of the detours around a reallocation points.

If it is impossible or prohibitively expensive to select primary and backup paths that satisfy the delay conditions needed for rerouting, or if the end-to-end delay of the

primary path or the backup paths is too great to be usable by the connection, then this channel is assigned a channel which is not re-routable.

4.3 Pre-allocation of Backup Channels

Since no a priori knowledge is available regarding the sequence of call setup requests and the associated bandwidth requirements, it is difficult to apply a deterministic technique for creating backup channels for primary subpath detours. A naive approach would be to create backup channels for all channels admitted to the network. However, this would require the network to allocate twice the resources, or more, for a channel than if no backup channels had been created at all.

To prevent explosive over allocation of resources for this technique, backup channels will only be preallocated, i.e. established in advance, for those channels most likely to be selected as victims for bandwidth reallocation. In addition, some backup channels can be multiplexed across their overlapping subpaths, and if the network determines that too many backup channels are being established, then the network may opt to establish fewer backup channels which cover more potential reallocation points each. However, each detour must still meet the QoS guarantees and transmission deadlines of each of the subpaths it is intended to replace.

The technique applied here is to allocate backup channels for a small percentage of the primary channels at a given node. The algorithm in Figure 2 is used to determine whether or not a channel needs pre-allocated backup channels for its local detours. The algorithm determines the set, **LP**, of channels that would need pre-allocated backup channels, selected from lowest priority up, until enough bandwidth, the minimum backup channel bandwidth threshold (MBBT), has been selected so that new high priority calls will not be delayed excessively during call setup. For each node, the set **LP** is recomputed each time a new channel is admitted. If the new channel ends up in the node's set **LP**, then the node will cause a backup channel to be pre-allocated, i.e. established, along the nodes detour. The MBBT can be dynamically adjusted by the network based upon its load and needs so that enough resources can be maintained in reserve for admitting high priority connections to the network.

4.4 Port Level Detours

In [25, 26], all channel reroutes were performed using backup channels that completely bypass the reallocation point node. In this paper, a finer grained approach is also studied. When a potential reallocation point notices that it is also a detour point for another node, and that a backup channel (of which it is an end-point) was pre-allocated for the other node, then it considers the possibility of using port-level detours whenever possible, for rerouting the channel locally. Thus, when a reallocation point needs to select a victim channel for bandwidth reallocation and channel rerouting, it checks to see if it is the end-point of a backup channel established for the channel at another node in the network. If so, and the backup channel does not flow through the output port(s) which are commandeering bandwidth from this channel, then the backup channel flowing out of this node can be

used, rather than using the backup channel specifically pre-allocated for this node, which bypasses it altogether. This works as long as the path length of the backup channel for which this node is an end-point does not significantly exceed the path length from this reallocation point to either of its detour points. This will reduce some of the complexity of the channel reroute, since only one reroute signal will need to be sent, and might reduce the channel reroute time since the reroute signal may not have to travel as far to reach the other end-point of the backup channel. Not every

Algorithm: CAC-Prealloc-Backup-Channels

Input: lc_m , NBC, PBC, BC

Output: NBC, PBC, BC

// lc_m is the channel being established at this node

// NBC is the set of channels for which no backup channel is preallocated

// PBC is the set of channels for which a backup channel might be preallocated

// BC is the set of channels for which a backup channel is definitely preallocated

If lc_m can be dropped or suspended indefinitely, **then**

 Add channel lc_m to the non-backed-up channel set **NBC**

 Do not add channel to the potential-backup set **PBC** and

 Do not add channel to the backedup channel set **BC**,

 Do not create backup channels for the local detour around this node

return

ElseIf channel lc_m can tolerate path computation and channel setup delays, **then**

 Add channel lc_m to the non-backed-up channel set **NBC**

 Do not add channel to the potential-backup set **PBC** and

 Do not add channel to the backedup channel set **BC**,

 Do not create backup channels for the local detour around this node

return

Else

 Add channel lc_m to the potential backup set **PBC**

 Let set **LP** contain channels lc_i which are the n lowest priority channels in **PBC** where:

$$n \geq k \text{ such that } \sum_{i=1}^k lc_i \geq \text{MBBT} \quad (\text{the Minimum Bandwidth Backup Threshold})$$

End-If

If $lc_m \in \text{LP}$ **Then** // i.e., channel lc_m is one of the channels to create a backup channel for
 build instructions for detour points p_1 and p_2 , the end points of the local detour around

 this node, to establish the backup channel along the specified local detour

 insert these instructions into the connection setup message

 forward the connection setup message (it will be processed by one of the detour points)

 wait for the call accept signal to come back

 when the call accept signal comes back, do the following:

For all channels $lc_k \notin \text{LP}$, such that $lc_k \in \text{PBC}$ **Do**

If lc_k has a pre-allocated backup channel along its local detour around this node **Then**
 release the backup channel since it is no longer needed

End-If

End-For

Else

 we do not need to create a backup channel along the local detour around this node

End-If

Fig. 1. Backup Channel Preallocation Setup Algorithm

channel reroute will be able to take advantage of port-level rerouting. However, whenever possible, the network will use this finer grained rerouting technique of bypassing the output port, rather than the entire reallocation point, since it will be quicker than its node-level counterpart.

5 Simulation Results

A cell-based discrete event simulator, called **SANRoP** (Simulator for ATM Network Routing Protocols), was developed that is capable of simulating the process of sending cells through an ATM switching network. Discrete events are executed to transfer cells and signals among the components of an ATM network. **SANRoP** uses the algorithms and procedures described here, and in [15, 16], for computing a channel’s primary path, primary subpaths, and local detours for backup channels.

The results of the simulation are presented in table 1. Two types of networks were simulated: mesh grid networks, and general topology networks. The mesh grid network consisted of 24 switches and 12 hosts. The general topology network consisted of 10 switches and 6 hosts. The network load varied from light (i.e., 0 – 25% overall average network utilization) to moderate (i.e., 25 – 260% overall average network utilization) during the times at which bandwidth reallocation and channel rerouting occurred. The cost adjustments shown represent the effects of using a detour that increases (or decreases in the case of sub-optimal primary paths with end-to-end detours) the overall cost of the path used by the channel. Small changes in cost represent a change of approximately 0 - 7.5% of the original primary path. Medium changes in cost represent a change of approximately 7.5 - 15% of the original primary path. Large changes in cost represent a change of approximately 15 - 30% of the original primary path. Very large changes in cost represent a change of approximately 30 - 60% of the original primary path. As the path lengths increased beyond the medium category the amount of buffering required by the switches during channel rerouting increased only when sub-optimal routing with end-to-end detours was used. However, in all cases, the maximum buffer size remained less than 20 cells, which can be easily handled with sufficient memory in the switches. In all simulation runs, no cells missed their transmission deadlines during channel reroute

Table 1. Simulation Results

Mesh-Grid Network									General Network					
			Sub-optimal / End-to-End		Optimal / End-to-End		Optimal / Node-Level		Sub-optimal / End-to-End		Optimal / End-to-End		Optimal / Node-Level	
# of re-routes	Cost Adj.	Net-work Load (% util)	Max Total Re-route Time	Max Buff. Range (cells)	Max Total Re-route Time	Max Buff. Range (cells)	Max Total Re-route Time	Max Buff. Range (cells)	Max Total Re-route Time	Max Buff. Range (cells)	Max Total Re-route Time	Max Buff. Range (cells)	Max Total Re-route Time	Max Buff. Range (cells)
2	Sm.	0-25	414.0	0-1	509.5	0-1	166.7	0-1	757.7	0-1	1163.6	0-1	249.7	0-2
4	Sm.	25-60	549.2	0-3	816.0	0-1	230.1	0-1	1406.9	0-2	1748.0	0-2	380.4	0-3
2	Med.	0-25	425.7	0-1	458.6	0	197.6	0	1077.6	0-3	1684.8	0	289.9	0
4	Med.	25-60	693.2	0-3	589.8	0	272.1	0-1	1267.6	6-9	2684.1	0	582.4	1-2
2	Lg.	0-25	601.7	4-9	749.2	0	228.9	0	1235.2	6-11	1819.8	0	488.4	0
4	Lg.	25-60	1018.6	6-8	1252.8	0	370.1	0-1	1522.9	8-15	2635.6	0	753.1	0-1
2	vLg.	0-25	941.3	11-14	1172.8	0	323.6	0	2195.7	12-18	2548.2	0	644.3	0

times, and the network fulfilled its QoS guarantees, even when bandwidth reallocation and channel rerouting occurred due to admission of a high priority channel for which one or more nodes had insufficient resources available.

The channel reroute times varied from 167 μ sec to 2,684 μ sec. This is well within the tolerances allowed with call establishment deadlines on the order of 2-5 sec. When an optimal path was used for the primary channel, the amount of buffering required remained fairly minimal, which is expected when detours that extend the cost of the channel are utilized. Also, the time required, and therefore the delay encountered by high priority channels requesting access to the network, actually goes down when node-level detours are used instead of end-to-end detours. Thus, when a victim is itself a real-time channel (albeit one with a relatively lower priority), it is likely to be rerouted fairly quickly and efficiently.

When a sub-optimal path was used for the high priority channels, the amount of buffering required depended upon several factors. First, as the change in the cost of the channel is increased (i.e., the path cost is reduced by larger amounts, since these channels are for real-time channels), the maximum required buffer space increased. However, as the link utilizations increased along the routes for the backup channels, the cells traversing the shorter route were sometimes slowed down, thus reducing the amount of buffering required by a minor amount. However, buffer sizes never grew very large, and in all cases, the channel reroute times were still quite acceptable.

The use of port level detouring allowed switches to select victims which did not have pre-allocated backup channels bypassing that switch. The switch simply used a backup channel allocated to bypass some other node in the network. However, this allowed nodes to continue to select victims, without needing to establish a new backup channel, under certain conditions when there were insufficient victims locally. This provided additional benefit to the nodes in about 10% of the cases where bandwidth reallocation and channel rerouting were required.

These results demonstrate that the QoS requirements of a channel can still be met and the ordering of the ATM cells to the application layer preserved when channel rerouting occurs, even when moderate traffic loads are present on the network. These results were expected due to the nature of the path selection algorithms. This means that, even for real-time channels, the cell stream integrity was maintained, i.e. the amount of delay due to route detouring was negligible and therefore insufficient to cause loss of QoS, which is an important requirement for real-time communications.

6 Conclusion

In this paper, bandwidth reallocation was examined as a tool for admission of high priority real-time channels in an ATM network which is experiencing partial overload. In a traditional ATM network, high priority calls may not be admissible due to a lack of available resources, since admitting them could potentially cause the QoS requirements of previously admitted calls to be violated. In order to admit them to the network, it is necessary to make sufficient resources available for the high priority channels while still meeting the QoS guarantees of the already admitted connections. When a high priority application requests a channel, bandwidth assigned to a lower priority channel is reallocated to the higher priority channel being established. Since the lower priority channel no longer has any bandwidth, it was

necessary to reroute the channel along a local detour around the node that caused the reallocation so that its own QoS requirements could still be met.

Routing issues for supporting end-to-end, local, and port-level detours were examined, and methods were given so that the paths used by a detour could meet the channel transmission deadlines. Optimal and sub-optimal routes were chosen for the primary path of the channel depending upon the priority level of the connection. Detours were selected which allowed the channel to continue to meet its QoS requirements when it was rerouted around a node that reallocated its bandwidth elsewhere. Methods were presented to ensure that the cells of a channel are delivered in sequence, and on time, to the destination, even though the cell stream was split in the middle and routed across two separate paths through the network. A simulation was constructed to demonstrate that the use of these detours resulted in the QoS guarantees and transmission deadlines still being met for rerouted real-time channels.

Future work will involve looking into the possibilities of multiplexing the backup channels for different primary channels along common subpaths so that fewer network resources, i.e. bandwidth, needs to be allocated to the backup channels themselves. This would increase the number of connections that can be accepted simultaneously by the network. The methods described in [8, 9] may be able to be adapted for use in the simulator in order to multiplex the backup channels assigned to be shared by the local detours into a single backup channel which requires less network resources. In addition, it would be nice to examine alternative methods for calculating a channel's primary path, and detour subpaths, which may lead to even more flexibility in selecting detours around a node which must reallocate the bandwidth of one channel to another. Finally, simulations need to be run for large heavily loaded networks with wildly varying traffic patterns as a function of time in order to see what happens in the worst case when a set of channel reroutes are performed simultaneously.

References

1. J. Anderson, B. Doshi, S. Dravida, and P. Harshavadhana, "Fast restoration of ATM networks," *IEEE Journ. on Selected Areas in Communications*, vol. 12, no. 1, pp. 128–138, January 1994.
2. J. Baker, "A distributed link restoration algorithm with robust preplanning," *Proc. IEEE GLOBECOM*, pp. 306–311, 1991.
3. A. Banerjea, C. Paris, and D. Ferrari, "Recovering guaranteed performance service connections from single and multiple faults," Technical Report TR-93-066, UC Berkeley, 1993.
4. A. Banerjea, "Simulation study of the capacity effects of dispersity routing for fault tolerant real-time channels," *Proc. ACM SIGCOMM*, pp. 194–205, August 1996.
5. I. Cidon, and R. Rom, "Multi-Path Routing combined with Resource Reservation", *IEEE INFOCOMM 97*, pp. 92-100, 1997.
6. W. Glover, "The selfhealing network: A fast distributed restoration technique for networks using digital crossconnect machines," *Proc. IEEE GLOBECOM*, pp. 1090–1095, 1987.

7. R. A. Guerin, A. Orda, "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms", *IEEE Trans. on Networking*, vol. 7, No. 3, pp. 350-364, 1999.
8. S. Han, and K.G. Shin, "Efficient Spare-Resource Allocation for Fast Restoration of Real-Time Channels from Network Component Failures," *Proc. IEEE RTSS 97*, pp. 99-108, 1997.
9. S. Han and K.G. Shin, "Fast Restoration of Real-Time communication Service from Component Failures in Multi-hop Networks," *Proc. ACM SIGCOMM*, pp. 77-88, 1997.
10. R. Iraschko, M. MacGregor, and W. Grover, "Optimal capacity placement for path restoration in mesh survivable networks," *Proc. IEEE ICC*, pp. 1568-1574, 1996.
11. R. R. Iraschko, M.H. MacGregor, and W. D. Grover, "Optimal Capacity Placement for Path Restoration in STM or ATM Mesh-Survivable Networks", *IEEE Trans. on Networking*, vol. 6, no. 3, pp.325-336 1998.
12. B. Kao, H. Garcia-Molina, and D. Barbara, "Aggressive transmissions of short messages over redundant paths," *IEEE Trans. Parallel and Distributed Systems*, vol. 5, no. 1, pp. 102-109, January 1994.
13. R. Kawamura, K. Sato, and I. Tokizawa, "Self-healing ATM networks based upon virtual path concept," *IEEE Journ. on Selected Areas in Communications*, vol. 12, no. 1, pp. 120-127, January 1994.
14. D. H. Lorenz, A. Orda, "QoS Routing in Networks with Uncertain Parameters", *IEEE Trans. on Networking*, vol. 7, No. 3, pp. 768-778, 1999.
15. L. Miller, and A. M. K. Cheng, "CAC with Bandwidth Reallocation and Active Channel Rerouting Techniques in ATM Networks: Simulation and Performance Results," *IASTED Applied Simulation &Modelling 2000*, pp. 184-190, 2000.
16. L. Miller, and A. M. K. Cheng, "Admission of High Priority Real-Time Calls in an ATM Network via Bandwidth Reallocation and Dynamic Rerouting of Active Channels", *IEEE RTSS 2000*, pp. 249-258, 2000.
17. K. Murakami, and H. Kim, "Near-optimal virtual path routing for survivable ATM networks," *Proc. IEEE INFOCOMM*, pp. 208-215, 1994.
18. A. Orda, "Routing with End-to-End QoS Guarantees in Broadband Networks", *IEEE Trans. on Networking*, vol. 7, No. 3, pp. 365-374, 1999.
19. P. Ramanathan, and K.G. Shin, "Delivery of time-critical messages using a multiple copy approach," *ACM Trans. Computer Systems*, vol. 10, no. 2, pp. 144-166, May 1992.
20. R. Vogel, R. Herrtwich, W. Kalfa, H. Wittig, and L. Wolf, "QoS-Based routing on multimedia streams in computer networks," *IEEE Journ. on Selected Areas in Communications*, vol. 14, no. 7, pp. 1235-1244, Sept. 1996.
21. C. Yang, and S. Hasegawa, "FITNESS: Failure immunization technology for network service survivability," *Proc. IEEE GLOBECOM*, pp. 1549-1554, 1988.
22. Q. Zheng, and K.G. Shin, "Fault-tolerant real-time communication in distributed systems," *Proc. IEEE FTCS*, pp. 86-93, 1992.

Temporal Control Specifications and Mechanisms for Multimedia Multicast Communication Services

Hsu-Yang Kung¹ and Chung-Ming Huang²

¹ Dept. of Management Information Systems, National Pingtung University of
Science and Technology, Pingtung, Taiwan, R.O.C.

`kung@mail.npust.edu.tw`

² Dept. of Computer Science and Information Engineering, National Cheng Kung
University, Tainan, Taiwan, R.O.C.

`huangcm@locust.iie.ncku.edu.tw`

Abstract. The Internet explosion impels the extensive demands for distributed multimedia presentations (DMPs), which provide multiple users with QoS-controlled multimedia services under multicast communications, such as media distribution and virtual classroom. In this paper, we (i) identify the primary issues for ensuring smooth multiple-stream multimedia presentations in the multicast environment, (ii) propose a formal temporal definition to specify related attributes of a multimedia presentation, and (iii) propose a temporal control mechanism to achieve the temporal synchronization of multicast multimedia presentations. Based on the proposed temporal definition and control mechanism, a multimedia middleware for multicasting multiple streams is developed. The multimedia middleware is named TVMS (The Visual Multicast Systems), which (i) provides a flexible authoring tool to allow users to author a multiple-stream multimedia presentation in a multicast environment and (ii) achieves smooth multimedia presentations.

1 Introduction

With the advances of computer and communication technologies, distributed multimedia presentations (DMPs), e.g., video distribution and virtual classroom, become popular applications [3,6,16]. DMPs can be characterized by the integrated multicast communications and presentation of multiple continuous and static media. Based on multicast communications, the server transmits data to multiple recipients simultaneously, each of who has the same multicast address [5]. A continuous medium, such as video or audio, is a time-dependent medium that possesses temporal relations between media units [2]. A static medium, such as text or still image, is a time-independent medium that has no temporal relation between media units, but may have inter-media temporal relations with other media streams. Since the transmission delay is undeterministic in

a distributed environment, temporal anomalies always exist during a multimedia presentation [11]. Thus, one of the important issues in implementing DMPs is to resolve the multiple-stream multimedia synchronization problem that is associated with the multicast delivery [13].

The goal of multiple-stream multimedia synchronization is to keep temporal relations of media streams as much as possible during the presentation. Multiple streams mean that media streams are retrieved from their own media bases and are transmitted via independent network channels with different network QoS (Quality-of-Service) requirements [12,17]. Therefore, the delay variance of multicast communications and the respective features of multiple streams complicate the multimedia synchronization reams complicate the multimedia synchronization [10,14].

Multimedia synchronization is one of the important issues for deriving smooth multimedia presentations in the distributed environment [1,8,18]. Two types of temporal synchronization are intra-medium synchronization and inter-media synchronization [11]. Intra-medium synchronization ensures intra-medium temporal relation of a medium stream and compensates for jitter, which is the asynchronous anomaly between consecutive media units of a medium stream [15]. Two widely adopted intra-medium synchronization schemes are blocking and non-blocking synchronization schemes [19]. (1) The blocking scheme: If an expected medium unit does not arrive on time, the presentation process suspends its presentation until the expected medium unit arrives. (2) The non-blocking scheme: If an expected medium unit does not arrive on time, the presentation process immediately re-presents the most recently received medium unit. Inter-media synchronization ensures inter-media temporal relations among related media streams and compensates for skew, which is the time difference between related media streams [15]. The master-stream control scheme can be adopted to achieve inter-media synchronization. The master stream dominates the commencement and finish of a presentation at each synchronization point. (1) If a slave stream finishes its presentation earlier than the master stream at a synchronization point, the slave stream has to block its presentation until the master stream finishes its presentation. (2) When the master stream finishes its presentation at a synchronization point, the late slave streams have to discard media units to keep pace with the master stream.

A DMP essentially consists of two components: an authoring system and a temporal control system [4,9]. An authoring system is the generating mechanism of behavior specifications. A behavior specification is composed of media attributes of related streams, including involved media streams, temporal and spatial attributes of each medium stream, and the temporal relationship between related media streams. An authoring system allows a user to specify behavior specifications of the corresponding multimedia presentation. The temporal control system is the synchronization and presentation mechanism that is derived from function specifications. Function specifications describe how to achieve temporal relations existing in the corresponding behavior specifications. In this paper, to resolve multimedia synchronization in the multicast presen-

tation environment, we propose a formal definition approach to have concise behavior specifications. Based on the proposed formal definition, a multicast multimedia middleware named TVMS (The Visual Multicast Systems) is developed to achieve smooth pre-orchestrated DMPs.

The rest of the paper is organized as follows. Section 2 describes the proposed multicast multimedia network architecture. Section 3 describes the formal definition of the temporal relationships in a multimedia presentation. Section 4 describes the system architecture of the temporal control mechanism. Section 5 concludes this paper.

2 Network Architecture

The proposed multicast multimedia network, which is called Multicast MultiMedia Communication Network (M^3CN), is a two level hierarchical architecture that spans a distributed environment [7]. The M^3CN consists of a WAN and a lot of LANs that are attached with the WAN. Each LAN is composed of a local Multicast MultiMedia Server (M^3 server) and clients. An M^3 server transmits media units to hosts via LAN or (and) WAN. Clients of a presentation group present the same multimedia resource simultaneously and maybe scattered over different LANs.

In M^3CN , the concept of a “virtual server” is adopted. A virtual server receives media units from the “physical server”, which owns the presentation resource, and re-transmits them to end clients. The virtual server is a local server of a LAN and compensates for WAN’s anomalies by means of pre-depositing some media units and having corresponding synchronization schemes. The concept of virtual servers can simplify the overhead of synchronization control in clients because WAN’s asynchronous anomalies are compensated for and media streams are synchronized at virtual servers. Clients become simpler and low-end, e.g. a Set-Top-Box, a diskless networking PC, or a networking TV.

The multicast presentation system is composed of the physical server system (PSS), the virtual server system (VSS) and the client system (CS). The PSS retrieves media units from media bases, and multicasts media units to virtual servers according to the information schedule file. Through the PSS, a system manager can specify a communication configuration that contains the multicast group address, communication socket ports, and media files. The VSS receives media units from the PSS and stores media units in media buffers temporarily. According to the presentation schedule, the VSS re-multicasts media units to clients with the synchronization control. With the help of the VSS, temporal anomalies induced by the WAN’s transmission can be compensated. The CS receives media units and achieves a smooth presentation adopting the synchronization and presentation control.

3 The Formal Definition of Multimedia Presentations

A formal definition mechanism should be proposed to specify related temporal attributes in a multimedia presentation. An authoring system can adopt the formal definition mechanism to have users to specify what their multimedia presentations look like. In this Section, we have the formal definition of a multimedia presentation.

A multimedia presentation MP is defined as $MP = \{MS, TS\}$, where $MS = \{ms_1, ms_2, \dots, ms_n\}$ represents the set of n involved media streams and $TS = \{p\text{-stage}_1, p\text{-stage}_2, \dots, p\text{-stage}_k\}$ represents the presentation temporal schedule. A presentation temporal schedule consists of many presentation *stages* ($p\text{-stage}_i$, where $i=1..k$). Each $p\text{-stage}_i$ contains x_i presentation *sections* ($p\text{-section}_i$, where $i=1..m$), where $1 \leq x_i \leq m$ and $x_1 + x_2 + \dots + x_k = m$. In other words,

$$\begin{aligned} p\text{-stage}_1 &= \{p\text{-section}_1, p\text{-section}_2, \dots, p\text{-section}_{x_1}\}, \\ p\text{-stage}_2 &= \{p\text{-section}_{x_1+1}, p\text{-section}_{x_1+2}, \dots, p\text{-section}_{x_1+x_2}\}, \\ &\dots \end{aligned}$$

A presentation stage is a semantic cut of a multimedia presentation. For example, let the multimedia presentation be CNN news broadcast about the chess race between world chess champion Gary Kasparov and supercomputer Deep Blue. Figure 1 depicts the presentation as follows. (1) The news reporter reports the news about a chess race between Gary Kasparov and Deep Blue. The news reporter's audio, Gary Kasparov's video, and the related news text are presented. (2) Gary Kasparov thinks and moves a piece. Then, the video of chess explanation and the text about the introduction of Gary Kasparov are presented. (3) An agent moves the piece according to Deep Blue's determination. The background music and some auxiliary texts are always presented. Thus, the presentation of Figure 1 is divided into three stages.

Media streams involved in a presentation stage are a subset of MS defined as $A\text{-}MS(p\text{-stage}_i) = \{A_i(ms_1), \dots, A_i(ms_n)\}$. In stage i , if medium stream ms_x has media units presented, $A_i(ms_x)=1$, $1 \leq x \leq n$; if medium stream ms_x has no media units presented, $A_i(ms_x)=0$, $1 \leq x \leq n$.

A presentation section represents that some media objects have temporal relationships, e.g. the start relation. Thus, one medium object's presentation in a section depends on another medium object's presentation status. For example, the text of news and the video of Gary Kasparov appear when a specific audio is presented. As depicted in Figure 1, the presentation of a specific segment A1 starts the presentations of V1 and T1.

It is not necessary that a medium stream always has media units presented throughout an entire section. In Figure 1, the text stream has nothing to present between t_2 and t_3 . The time period that a medium stream has nothing to present is called an *idle segment*. Idle segments can be defined as $i\text{-segment}(p\text{-stage}_i, ms_j) = [t_x, t_y]$, where $A_i(ms_j)=1$ and ms_j has nothing to present from time point t_x to time point t_y . For example, the text medium has an idle segment δ_1 from t_2 to t_3 in section 2 of stage 1.

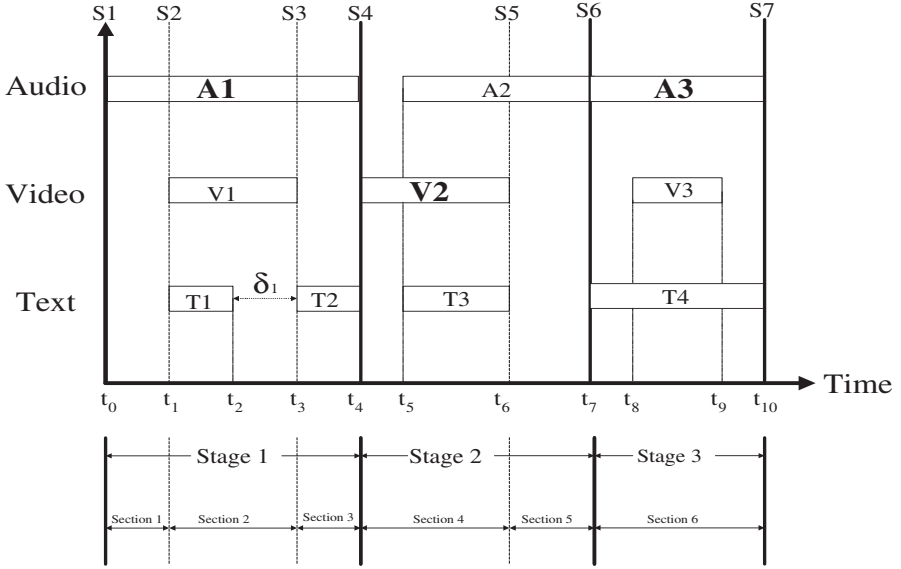


Fig. 1. The display time-bar chart of a presentation temporal schedule.

Temporal relationships of two media objects can be formally defined as $x(t_1) \oplus y(t_2) [t_3, t_4]$, where x and y denote media objects, t_1 is the display time period of x , t_2 is the display time period of y , $\oplus$ denotes the type of the temporal relationship, and t_3 and t_4 describe the front, tail, or gap time interval for the corresponding temporal relationship. (Parameters t_3 and t_4 are optional. If there is no front, tail, or gap time period, nothing has to be specified.) The temporal relations denoted by $\oplus$ include the ‘equal’, ‘start’, ‘before’, ‘meet’, ‘during’, ‘overlap’, ‘finish’, and their reversed relations.

Based on the proposed formal definition, temporal relationships for involved media objects in a presentation stage is defined as $TR(p\text{-stage}_i) = \{O_{P_{i1}}(t_{P_{i1}}) \oplus O_{Q_{i1}}(t_{Q_{i1}}) [x t_{i1}, y t_{i1}], \dots, O_{P_{iI}}(t_{P_{iI}}) \oplus O_{Q_{iI}}(t_{Q_{iI}}) [x t_{iI}, y t_{iI}]\}$, where media objects $O_{P_{ij}}$ and $O_{Q_{ij}}$, $1 \leq j \leq I$, has some temporal relationship in stage i . Each presentation stage is associated with a master stream. The master stream of a stage is formally defined as $M(p\text{-stage}_i) = \{ms_y\}$, where $1 \leq y \leq n$ and $A_i(ms_y) = 1$. Figure 2 is the formal definitions of the illustrated multimedia presentation depicted in Figure 1

4 Software Architecture of the Temporal Control Mechanism

This Section describes the system architecture and prototype implementation of TVMS, which is composed the physical server system (PSS), the virtual server system (VSS), and the client system (CS).


```

MP = {MS, TS},
MS = {video, audio, text},
TS = {stage-1, stage-2, stage-3},
stage-1 = {section-1, section-2, section-3},
stage-2 = {section-4, section-5},
stage-3 = {section-6},
A-MS(stage-1) = {1, 1, 1},
A-MS(stage-2) = {1, 1, 1},
A-MS(stage-3) = {1, 1, 1},
i-segment(stage-1, video) = [t0, t1],
i-segment(stage-1, video) = [t3, t4],
i-segment(stage-1, text) = [t0, t1],
i-segment(stage-1, text) = [t2, t3],
i-segment(stage-2, audio) = [t4, t5],
i-segment(stage-2, video) = [t6, t7],
i-segment(stage-2, text) = [t4, t5],
i-segment(stage-2, text) = [t6, t7],
i-segment(stage-3, video) = [t7, t8],
i-segment(stage-3, video) = [t9, t10],
M(stage 1) = M(stage 3) = {audio},
M(stage 2) = {video},
TR(stage-1) = {A1(t4-t0) during V1(t3-t1)[t1-t0, t4-t3], A1(t4-t0) during T1(t2-t1)[t1-t0, t4-t2],
A1(t4-t0) finish T2(t4-t3)[t3-t0]},
TR(stage-2) = {V2(t6-t4) overlap A2(t7-t5)[t5-t4, t7-t6], V2(t6-t4) finish T3(t6-t5)[t5-t4]},
TR(stage-3) = {A3(t10-t7) during V3(t9-t8)[t8-t7, t10-t9], A3(t10-t7) equal T4(t10-t9)}.

```

Fig. 2. Formal definitions of the the illustrated multimedia presentation depicted in Figure 1.

4.1 Physical Server System (PSS)

Three main components of the PSS are Synchronizer, Media Sender, and Continuous Media Reader, which are depicted in Figure 3.

- Synchronizer. Synchronizer is responsible for the coarse-grain synchronization to achieve section and stage synchronization based on the parallel-last scheme. With the parallel-last scheme, each medium stream can be completely transmitted regardless of media processing anomalies.
- Media Sender. Media Sender is responsible for retrieving media units from media buffers, and then transmits them to networks. Moreover, the media sender should cooperate with Synchronizer to achieve inter-media synchronization.
- Continuous Media Reader. Continuous Media Reader is responsible for retrieving continuous media units from the media base and then puts them into media buffers. The purpose of media buffers is to compensate the irregular media retrieval time from the media base. For static media, since (1) the volume of media units is much less than that of continuous and (2) temporal requirement is not critical, static media units is directly retrieved from media bases by the corresponding Media Senders.

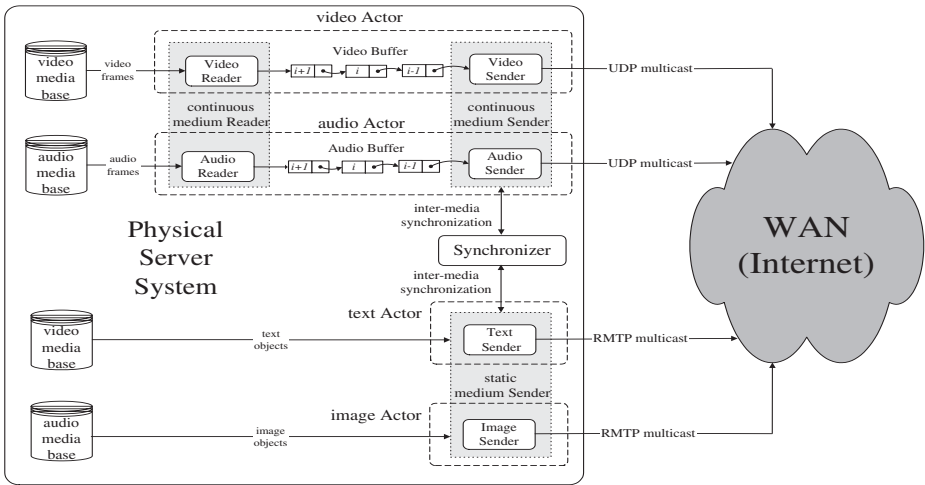


Fig. 3. Architecture of the physical server system.

Rate control is used to keep a continuous and steady multicasting for continuous media streams. For example, assume that the default transmission rate of a video stream is 15 frames-per-second (fps). Hence, Media Reader has to retrieve a video frame for every $1/15$ second from the video base and put the video frame into the video buffer. However, since a regular operating system, e.g., Unix and Windows NT, is a time-sharing and a multiple-process system. It is difficult to exactly control what time to retrieve a video frame and what time to accurately multicast a video frame. Due to the inaccuracy execution-time, Media Reader can not retrieve media units from the media base with a constant retrieving rate. As a result, the media buffer may become empty because Media Sender multicasts media units with the default transmitting rate. Under the situation of buffer empty, Media Sender has to suspend its work and then waits for Media Reader to retrieve media units into buffer. The suspending time induces a discontinuous transmission.

4.2 Virtual Server System (VSS)

Three main components of VSS are Media Receiver, Media Transmitter, and Synchronizer, which are depicted in Figure 4. These three components of VSS are similar to those in PSS. That is, functions of Media Receiver (Media Transmitter) in VSS are similar to those of Media Reader (Media Sender) in PSS. For simplicity, we only describe main features and functionality of VSS.

- Media Receiver. Because of the characteristics of static media, Static Media Receiver can not lose any medium unit. Thus, RMTP is used between the virtual servers and the physical server for static media. With the reliability function of RMTP, each static media unit can be received.

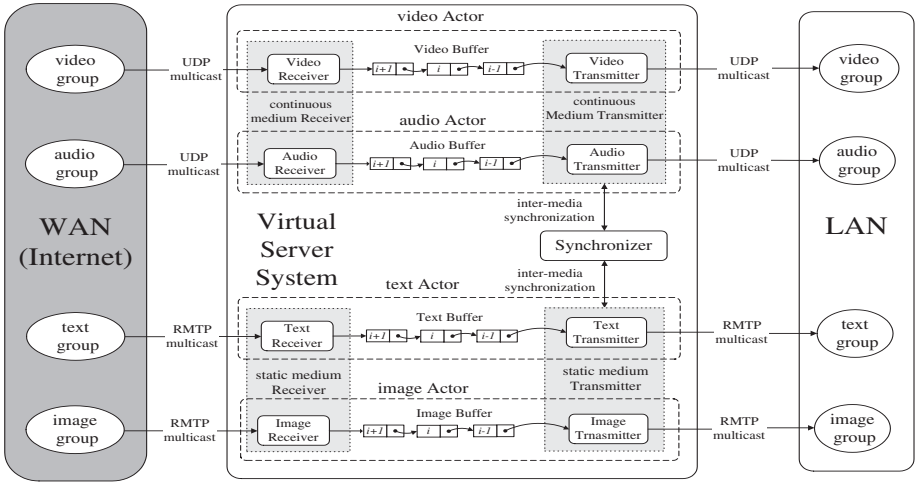


Fig. 4. Architecture of the virtual server system.

- Media Transmitter. Media Transmitter retrieves one medium unit from the media buffer and then re-multicasts the medium unit to LAN's clients according to the schedule description file. Two types of Media Transmitter are Continuous Media Transmitter and Static Media Transmitter, which are responsible for multicasting continuous and static media units respectively. RMTP is also used to transmit static media for the reason of reliability.
- Synchronizer. Synchronizer is also responsible for coarse-grain synchronization and to coordinate Transmitters' behavior.

In VSS, the master-medium-based synchronization control combining with the adopted presentation scheme, which is either the content-oriented or the time-oriented scheme, is executed at each synchronization point.

4.3 Client System (CS)

CS starts the presentation after some commencement control is done, i.e., after pre-depositing some media units in the buffer to compensate LAN's anomalies. Three main components of CS are Media Gather, Media Presenter, and Synchronizer, which are depicted in Figure 5. In CS, an additional function of Media Presenter is to achieve the fine-grain synchronization between continuous media under the condition that a tight temporal relation between audio and video streams, e.g., lip synchronization, is needed.

The master Media Presenter controls fine-grain synchronization by issuing fine-grain control messages to all of the other non-master Media Presenters at the fine synchronization point. The slower Media Presenters have to keep pace with the fastest Presenter by discarding some media units to reach fine-grain synchronization points. Synchronizer controls the coarse-grain synchronization between two consecutive sections. Furthermore, at the beginning of each section,

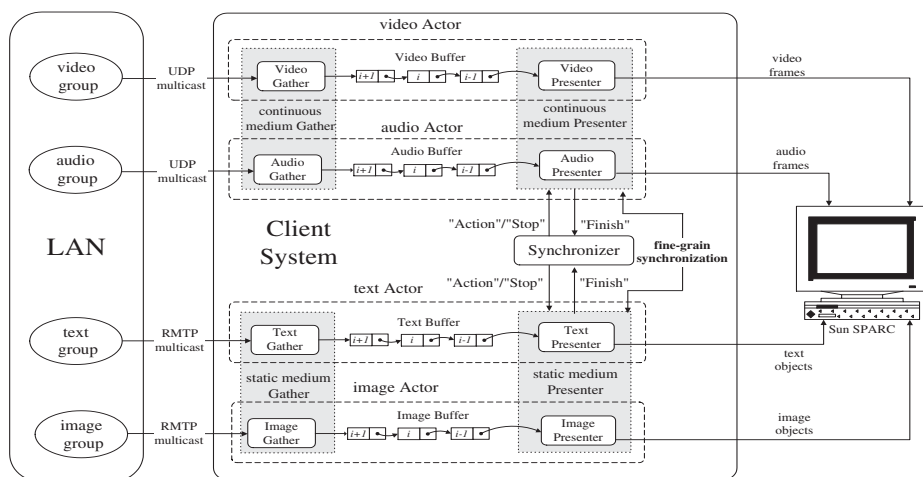


Fig. 5. Architecture of the client

Synchronizer has to assign one Presenter as the master and issues the fine-grain synchronization flag to the master Presenter in order to notify whether the fine-grain synchronization is needed or not in this section.

5 Conclusion

This paper describes the main issues of designing temporal control mechanisms for multicasting multiple-stream multimedia presentations. A formal definition is proposed for the behavior specification, which concisely specifies the seven temporal relationships of a DMP behavior. Based on the formal definition, we have developed the TVMS. TVMS is based on the proposed M^3CN architecture and contains synchronization/presentation mechanisms to achieve multiple-stream multimedia temporal control. TVMS also provides generic supports for media specifications and multicast environment setup. System developers can incorporate TVMS to develop DMPs more efficiently and effectively in a multicast environment.

References

1. Ali, Z., Ghafoor, A., Lee, C.S.G.: Media Synchronization in Multimedia Web Using a Neuro-Fuzzy Framework. *IEEE Journal on Selected Areas in Communications* **18** (2) (2000) 61–69
2. Blakowski, G., Steinmetz, R.: A Media Synchronization Survey: Reference Model, Specification, and Case Studies. *IEEE Journal on Selected Areas in Communications* **14** (1) (1996) 5–35

3. Bolot, J.C., Turetti, T.: Experience with Control Mechanisms for Packet Video in the Internet. *ACM SIGCOMM Computer Communication Review* **28** (1) (1998) 4–15
4. Coulson, G.: A Configurable Multimedia Middleware Platform. *IEEE MultiMedia Magazine*, **6** (1) (1999) 62–76
5. Deering, S.: Host Extensions for IP Multicasting. Request for Comments 1112, Network Working Group, Stanford University, (1989)
6. Diot, C., Dabbous, W., Crowcroft, J.: Multicast Communication: Survey of Protocols, Functions, and Mechanism. *IEEE Journal on Selected Areas in Communications* **15** (3) (1997) 277–290
7. Huang, C.M., Kung, H.Y.: A Synchronization Infrastructure for Multicast Multimedia at the Presentation Layer. *IEEE Transactions on Consumer Electronics*, **43** (3) (1997) 370–380
8. Huang, C.M., Lo, C.M.: An EFSM-based Multimedia Synchronization Model and the Authoring System. *IEEE Journal on Selected Areas in Communications* **14** (1) (1996) 138–152
9. Jourdan, M., Layaida, N., Roisin, C., Sabry-Ismail, L., Tardif, L.: Madeus and Authoring Environment for Interactive Multimedia Documents. *Proc. 6th ACM International Multimedia Conference (ACM Multimedia'98)*, Bristol, England, (1998) 267–272
10. Li, X. Ammar, M.H., Paul, S.: Video Multicast over the Internet. *IEEE Network* **13** (2) (1999) 46–60
11. Li, V.O.K. Liao, W.: Distributed Multimedia System. *Proceedings of IEEE* **85** (7) (1997) 110–195
12. Mathy, L., Edwards, C., Hutchison, D.: Principles of QoS in Group Communications. *Telecommunication Systems* **11** (1) (1999)
13. Panzieri F., Rocchetti, M.: Synchronization Support and Group-Membership Services for Reliable Distributed Multimedia Applications. *ACM/Springer-Verlag Multimedia Systems Journal* **5** (1) (1997) 1–22
14. Pasquale, J.C., Polyzos, G.C., Xylomenos, G.: The Multimedia Multicasting Problem. *ACM Multimedia Systems* **6** (1) (1998) 43–59
15. Steinmetz, R.: Human Perception of Jitter and Media Synchronization. *IEEE Journal on Selected Areas in Communications* **14** (1) (1996) 61–72
16. Willebeek-LeMair M.H., Shae, Z.: Videoconferencing over Packet-based Networks. *IEEE Journal on Selected Areas in Communications* **15** (6) (1997) 1101–1114
17. Xiao X., Ni, L.M.: Internet QoS: A Big Picture. *IEEE Network* **13** (2) (1999) 8–18
18. Yuang, M.C., Lo, B.C., Chen, Yu.G., Tien, P.L.: A Synchronization Paradigm with QoS Guarantees for Multimedia Communications. *Proc. Global Telecommunications Conference (GLOBECOM '99)*, (1999) 214–220
19. Zarros, P.N., Lee, M.J., Saadawi, T.: A Synchronization Algorithm for Distributed Multimedia Environments. *ACM Multimedia Systems* **4** (1) (1996) 1–11

Improving Fairness and Throughput in Multi-Hop Wireless Networks

Hung-Yun Hsieh and Raghupathy Sivakumar

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332, USA
{hyhsieh, siva}@ece.gatech.edu
<http://www.ece.gatech.edu/research/GNAN>

Abstract. In this paper we study the impact of the medium access control (MAC) layer and the routing layer on the performance of a multi-hop wireless network. At the medium access control layer, we argue that the notion of per-node fairness employed by the IEEE 802.11 standard is not suitable for a multi-hop wireless network where flows traverse multiple hops. We propose a new MAC protocol that supports prioritized per-node fairness and significantly improves performance in terms of both throughput and fairness. At the routing layer, we show that load balanced routing improves performance regardless of the nature of the underlying MAC protocol. Moreover, we show that an ideal load balanced routing protocol should take into account both the hop counts and the capacities when computing the optimal path. We propose a new routing protocol that improves performance over the conventional shortest-widest path routing.

1 Introduction

Ad-hoc networks are multi-hop wireless networks that lack the services of an established backbone infrastructure. They are typically formed by a collection of mobile stations cooperatively establishing a multi-hop wireless network. In recent years, numerous approaches have been proposed for routing [6,7,11-13], and medium access control (MAC) [1,4,10] in ad-hoc networks. While a majority of the routing protocols are similar to shortest path routing in that they use hop count as the optimization metric, the MAC schemes are mainly based on the CSMA/CA protocol. In this paper, we revisit the throughput and fairness properties of shortest path routing and CSMA/CA based MAC protocols in ad-hoc networks. We show through simulations that the end-to-end throughput and fairness properties of these routing and medium access control schemes are poor. We present simple algorithms at the two layers that significantly improve the throughput and fairness.

We make two key contributions in this paper: (i) We demonstrate that existing MAC protocols for ad-hoc networks (e.g. IEEE 802.11 [2]), based on the per-node fairness paradigm of CSMA/CA, do not provide end-to-end throughput fairness. We argue for a departure from the notion of per-node fairness to that

of per-flow fairness. *We then present a new MAC protocol that has a per-flow notion of fairness for channel access and achieves improved end-to-end throughput fairness.* (ii) We show that load balanced routing not only can improve the end-to-end throughput observed by flows, but also can have a positive impact on the fairness observed by flows. We argue that a conventional load balanced scheme such as shortest-widest path algorithm will not provide optimal results in ad-hoc networks. Finally, *we present a new load balanced routing algorithm that is suitable for the target environment.*

The rest of the paper is organized as follow: Section 2 presents the protocols and algorithms that we use in the rest of the paper. Section 3 describes the simulation model including the topology and traffic generation. Section 4 presents the simulation results. Section 5 discusses some issues and concludes the paper.

2 Algorithms

2.1 Medium Access Control

We use the IEEE 802.11 MAC protocol as the reference protocol. In order to alleviate any unfairness that the implementation of IEEE 802.11 protocol might contribute [8], we have implemented an ideal, per-node-fairness based MAC protocol (ILP) similar to the one presented in [9]. The ILP algorithm attempts to provide ideal, per-node fairness, and given a certain fairness level tries to maximize the throughput. Finally, we use an ideal per-flow-fairness based MAC protocol (IFP) that incorporates priorities in the ILP algorithm, where the priority of a node is set proportional to the number of flows traversing the node. Figure 1 presents a pseudo-code for the IFP protocol. Section 4 will present the simulation results comparing the three protocols.

2.2 Routing

We use a simple shortest path routing algorithm as the reference protocol. Initially, we show that the shortest-widest path algorithm is not suited to the ad-hoc network environment. For the rest of the simulations, we adopt a new load balanced routing algorithm that takes into account both the capacity (width) and the hop count (length) along a path. We assign a weight w to each “link” in the network, where w is proportional to the amount of contention at that link due to existing flows in the network. The shortest-widest path algorithm would then translate into finding the path with the minimum maximum-weight (MMW), while the new algorithm would involve finding the path with the minimum aggregate-weight (MAW). Figure 2 presents the algorithm for the MAW protocol. Note that a variation of Dijkstra’s algorithm (minimum maximum-weight instead of minimum aggregate-weight) can be used to achieve MMW routing with the same algorithm as shown in Figure 2. We show that the MAW algorithm performs better than the MMW algorithm in terms of the mean and variance of the end-to-end throughput. Finally, we demonstrate that the load balanced algorithm improves the fairness irrespective of whether the underlying MAC protocol is fair or unfair.

Input:

Set F of source-destination pairs (s_i, d_i)

Vector $Degree$

where $Degree(s_i)$ is the degree of node s_i

Vector $NumberOfFlows$

where $NumberOfFlows(s_i)$ is the number of flows traversing node s_i

Vector $Priority$

where $Priority(s_i)$ is the priority associated with node s_i

Vector $Allocation$

where $Allocation(s_i)$ is the number of time slots allocated to node s_i

(Both $Priority(k)$ and $Allocation(k)$ are set to 0 for all k during network initialization. The values carry over across iterations of the algorithm presented below.)

Output:

Set T of source-destination pairs allowed to transmit in the current time slot

Updated vector $Priority$

Updated vector $Allocation$

Algorithm:

Initialize set T to an empty set

While F is not empty

Find (s_i, d_i)

such that s_i has the maximum value in the lexicographic ordering of $(Priority(s_j), -Degree(s_j))$ for all s_j in F

Remove (s_i, d_i) from F

Add (s_i, d_i) to T

For each pair (s_j, d_j) in F

If node s_j is adjacent to node d_i

Remove pair (s_j, d_j) from F

If node d_j is adjacent to node s_i

Remove pair (s_j, d_j) from F

For each pair (s_i, d_i) in T

Increment $Allocation(s_i)$ by 1

$Priority(s_i) \leftarrow -Allocation(s_i)/NumberOfFlows(s_i)$

Fig. 1. *Ideal Per-Flow-Fairness Based MAC Protocol (IFP)*

3 Simulation Model

We use the *ns2* network simulator for our simulations [3]. While we have used topologies of varying sizes (50, 100, and 200 nodes respectively) for our simulations, we present only the results for the 100 node topology in this paper.

Input:Set F of source-destination pairs (s_i, d_i) **Output:**Set R of routes for all source-destinations pairs in F **Algorithm:**Initialize R to an empty setInitialize $weight(s_j)$ to 1 for all s_j For each pair (s_i, d_i) in F Use Dijkstra's shortest path algorithm to obtain route r_i For each node m on route r_i except for d_i Increment $weight(m)$ by 1 Increment $weight(q)$ by 1 for all q that is adjacent to m Insert r_i in R **Fig. 2.** *Load Balanced Routing*

The nodes are uniformly distributed in a 1500m x 1500m grid. The simulation scenarios presented in this paper do not have any mobility. We will revisit the issue of mobility later in Section 5. The data rate of the underlying channel is set to 2 Mbps, and the transmission range is set to 250m. The traffic in the network consists of 25 bi-directional TCP flows between 25 pairs of randomly (uniformly distributed) chosen sources and destinations. The simulations are run for a period of 100 seconds. Each data point is an average over 10 simulations run with different seeds for the random distribution. *We use the mean and the deviation as the metrics to compare the throughput and fairness respectively.* Unless otherwise specified, the routing protocol used is shortest path routing (SPR).

4 Simulations

4.1 MAC and Fairness

In Figure 3, we present the normalized deviation of the end-to-end throughput for the three MAC protocols. We define normalized deviation for a scenario as the standard deviation normalized by the mean throughput achieved for that scenario. As seen, IEEE 802.11 exhibits a high degree of unfairness. Note that in addition to the reasons given shortly, IEEE 802.11 has been shown to exhibit unfairness even when providing per-node fairness, and this accounts for the difference in its performance when compared to the ILP algorithm. The difference in performance between ILP and IFP can be briefly explained as follows: In ILP, nodes are given “equal” access to the channel irrespective of the number of flows traversing them. This results in lowered throughput for flows that traverse

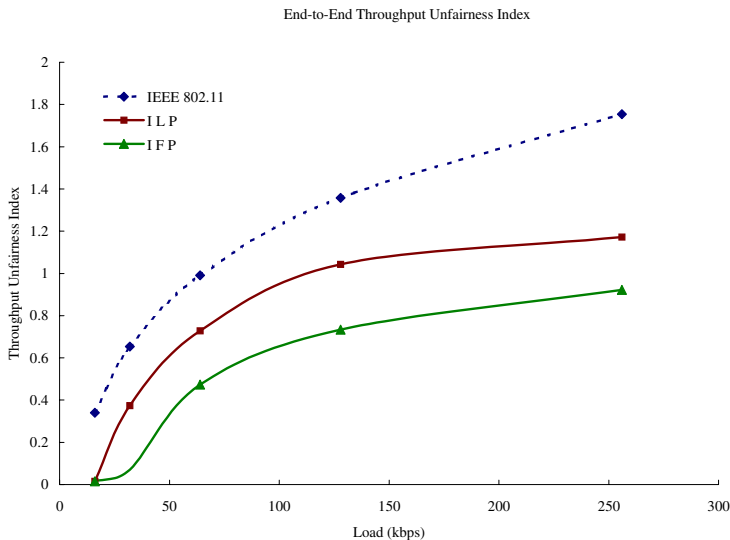


Fig. 3. MAC and Fairness

nodes handling more number of flows. However, in IFP, nodes are given access to the channel in proportion to the number of flows for which they act as relays (routers). Hence, flows are not penalized for traversing “congested” nodes. This results in the improved fairness for IFP.

4.2 Load Balanced Routing

In Figure 4, we present a comparison between the mean throughput achieved by the MMW (minimum maximum-weight, or shortest-widest path), and the MAW (minimum aggregate-weight) algorithms respectively. As observed, the MAW algorithm offers significantly more throughput than the MMW algorithm irrespective of the MAC protocol used. The reason behind the improvement is the fact that the network is moderately to heavily loaded (16 kbps to 256 kbps), and in such scenarios the longer hop counts (8.86 hops) of the MMW algorithm results in the network being overloaded sooner than in the case of the MAW algorithm (5.02 hops). Briefly, the larger number of hop counts results in more usage of the underlying network capacity:

$$Usage \approx NumberOfFlows * AverageHopCount * AverageFlowRate$$

As long as the total usage is less than the network capacity [5], the impact of larger hop counts is not noticed. However, when the network is heavily loaded, it is more likely that the larger hop count will result in the network becoming overloaded sooner, resulting in poor performance.

While the MAW algorithm is better in terms of the mean throughput, it can be seen from Figure 5 that the algorithm performs better in terms of the fairness

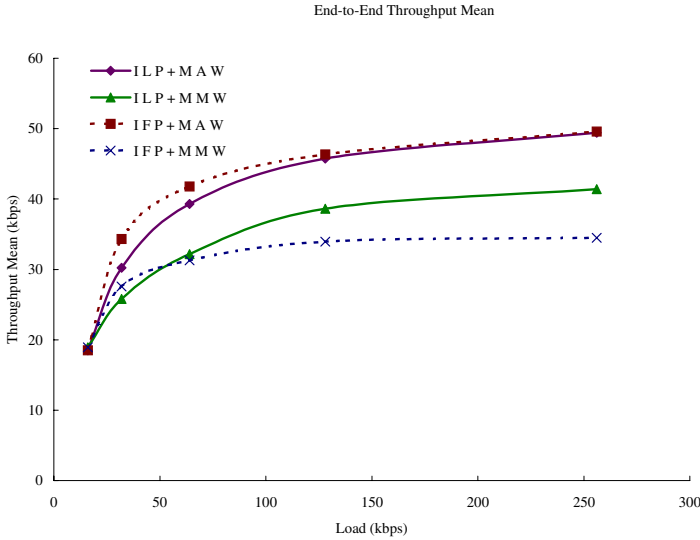


Fig. 4. Load Balanced Routing: Mean

also. Recall that the normalized deviation, and not the absolute deviation, is used as the fairness index.

4.3 Routing and Fairness

In Figure 6, we present the impact of the routing algorithm on the end-to-end throughput fairness. We again use the normalized deviation as the metric for fairness. When the underlying MAC is unfair, it is obvious that having a load balanced routing algorithm will improve fairness. This is because of the fact that load balancing reduces the average degree of multiplexing of flows on a single link, and hence bounds the unfairness introduced by the MAC protocol. This improvement in fairness is evident in Figure 6. However, it is interesting to note that *load balancing improves fairness even when the underlying MAC is fair with respect to flows*. Briefly, the reasons for this improvement are twofold: (i) The transport protocol used is TCP, and TCP is unfair to flows with larger RTTs. Hence, when flows with different RTTs share a single link, the mechanics of TCP will result in the flow with the smaller RTT getting a greater portion of the link capacity. Load balanced routing reduces the overlapping of flow paths, and hence reduces such effects. (ii) Although the underlying MAC protocol is fair, the variance in the degree of path overlapping (due to the existence of flows that have no or minimal link sharing along their paths, along-with flows that share links with a large number of flows) will induce unfairness in the network. Load balanced routing reduces the variance in the degree of path overlapping, and hence improves fairness.

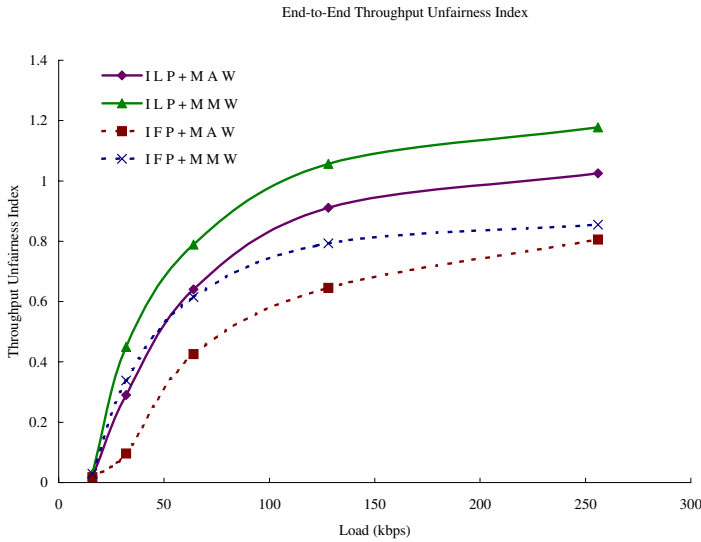


Fig. 5. Load Balanced Routing: Normalized Deviation

4.4 Routing and Throughput Distribution

In our simulations, we observe that shortest path routing occasionally exhibits higher average throughput than load balanced routing. While superficially this indicates better performance, a closer look at the average throughput distribution between the different flows reveal that shortest path routing, although exhibiting higher average throughput, punishes a large number of flows (very low throughput in relation to the mean) in favor of a few flows that enjoy throughputs significantly higher than the mean throughput. Figure 7 shows the distribution of the number of flows observing different end-to-end throughputs. The distribution is a consolidation of the results of 10 simulations, and hence has a total of 500 flows. As seen in the figure, the peak of the distribution for load balanced routing is closer to the mean than that of shortest path routing. Moreover, load balanced routing has a consistently better distribution curve about the mean throughput value. Finally, it can be seen that the peak of the distribution for the shortest path algorithm at the right end of the graph (high throughput) is higher than that of load balanced routing, substantiating our earlier claims that SPR greatly favors a few flows.

5 Issues and Summary

5.1 Issues

(i) *Mobility*: Due to lack of space, we do not consider mobility in the results presented thus far. However, the following observation can be made about the probable impact of mobility: While the shortest path and the MAW algorithms

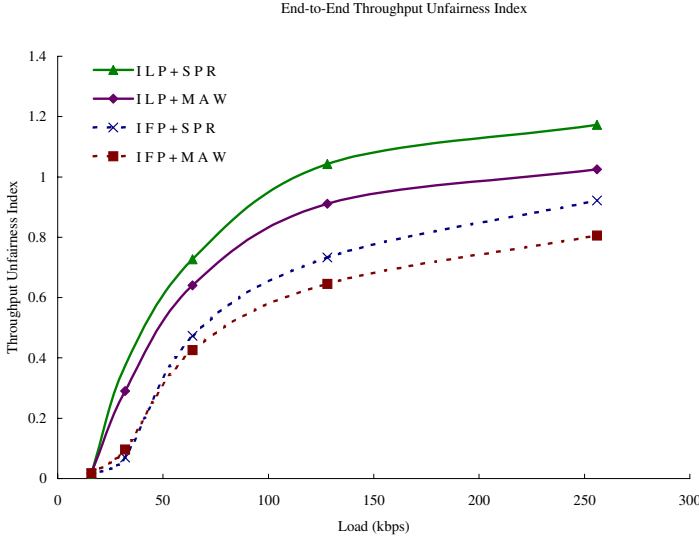


Fig. 6. Routing and Fairness

will suffer throughput degradation (possibly by the same amount) due to mobility induced losses, MMW can be expected to suffer significantly more losses. This is because of the fact that MMW paths, by virtue of their longer hop counts are more likely to break because of link failures. (ii) *Distributed Algorithms*: The new algorithms presented in this paper are centralized in nature. The scope of the paper is limited to highlighting the drawbacks of existing protocols and suggesting better approaches, and hence we do not present distributed versions of the algorithms. However, we believe that developing distributed versions of the algorithms introduced will not be a difficult task, and we hope to develop the distributed algorithms as part of our future work.

5.2 Summary

In this paper we have studied the performance of existing MAC and routing schemes in terms of their fairness and throughput characteristics. While we agree that the per-node fairness model adopted for packet cellular networks is apt for that environment, *we argue that such a model is not suitable for an ad-hoc network where nodes cooperatively act as routers or relays for flows belonging to other nodes in the network*. We propose a new MAC protocol that supports a per-flow fairness model, and in the process achieves significantly better end-to-end throughput fairness. At the routing layer, we show that *a load balanced routing scheme that takes into account both the capacity of paths and their hop counts is more suitable for ad-hoc networks* than a conventional shortest-widest approach. We demonstrate through simulations that the new routing algorithm does better than shortest path routing both in terms of throughput distribution and fairness.

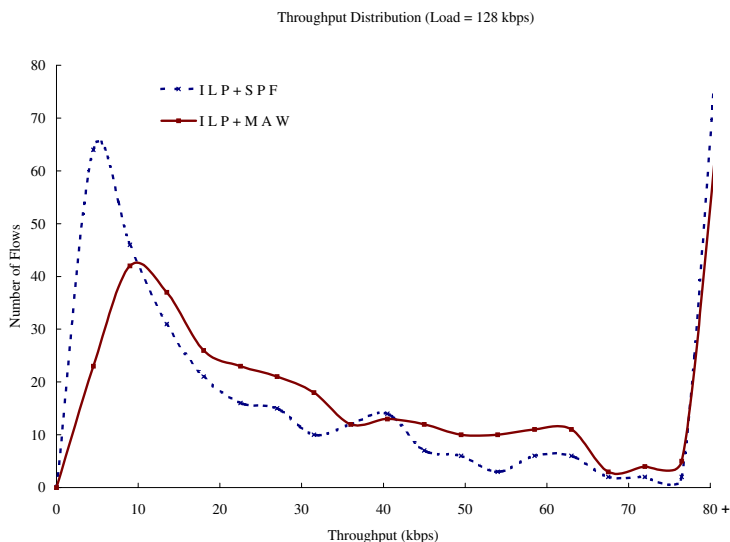


Fig. 7. Routing and Throughput Distribution

6 Acknowledgments

We thank the Yamacraw organization (<http://www.yamacraw.org>) for their generous support and funding part of this work.

References

1. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Medium Access Protocol for Wireless LANs," in *Proceedings of ACM SIGCOMM*, London, England, August 1994.
2. B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116-126, September 1997.
3. K. Fall and K. Varadhan, "*ns* Notes and Documentation," available from <http://www.isi.edu/nsnam/ns/>, 1999.
4. C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet Radio Networks," in *Proceedings of ACM SIGCOMM*, Cambridge, MA USA, August 1995.
5. P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388-404, March 2000.
6. Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *Internet Draft draft-zone-routing-protocol-01.txt*, August 1998.

7. D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, Chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996.
8. C. E. Koksal, H. I. Kassab, and H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," in *Proceedings of ACM SIGMETRICS*, Santa Clara, CA USA, June 2000.
9. H. Luo, S. Lu, and V. Bharghavan, "A New Model for Packet Scheduling in Multihop Wireless Networks," in *Proceedings of IEEE MOBICOM*, Boston, MA USA, August 2000.
10. J. P. Monks, V. Bharghavan, and W. W. Hwu, "Transmission Power Control for Multiple Access Wireless Packet Networks," in *Proceedings of IEEE LCN*, Tampa, FL USA, November 2000.
11. S. Murthy and J. J. Garcia-Luna-Aceves, "A Routing Protocol for Packet Radio Networks," in *Proceedings of ACM SIGCOMM*, Cannes, France, September 1997.
12. C. E. Perkins, E. M. Royer, and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing," *Internet Draft draft-ietf-manet-aodv-04.txt*, October 1999.
13. R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm," *IEEE Journal on Selected Areas in Communications (Special Issue on Ad-hoc Routing)*, vol. 17, no. 8, pp. 1454-1465, August 1999.

Dynamic Allocation of Transmitter Power in a DS-CDMA Cellular System Using Genetic Algorithms^{*}

Jie Zhou¹, Yoichi Shiraishi², Ushio Yamamoto¹, and Yoshikuni Onozato¹

Dept. of Computer Science, Faculty of Engineering, Gunma University
1-5-1 Tenjin-cho, Kiryu, Gunma, 376-8515 Japan

¹{zhou, onozato, kansuke}@nzt1.cs.gunma-u.ac.jp
~http://www.nzt1.cs.gunma-u.ac.jp

²siraishi@keim.cs.gunma-u.ac.jp
~http://www.keim.cs.gunma-u.ac.jp

Abstract. In this paper, we propose an approach to solve the power allocation issues in a DS-CDMA cellular system using genetic algorithms (GAs). The transmitter power control has proven to be an efficient method to control cochannel interference in cellular PCS, increase bandwidth utilization and balance the wireless services. Most of the previous studies have assumed that the transmitter power level is controlled in a constant domain under the assumption of uniform distribution of users in the coverage area or in a continuous domain. In this paper, the optimal Centralized Power Control (CPC) vector is characterized and its optimal solution for CPC is presented using GAs in a large scale DS-CDMA cellular system.

1 Introduction

Transmitter power control is an effective way of increasing the system capacity and transmission quality in cellular wireless systems. In DS-CDMA wireless communication systems, transmitter power is regulated to be provided to each user randomly distributed in coverage area (service zone), which is considered as the interference seen by other users. There has been significant work on the channel assignment and power allocation strategy used in these systems, such previous works as Refs.[1][2] which have focused on CPC. They only investigated simple case because of the difficulties of computation and search of optimal solutions.

In this paper, we first propose an approach to solve the power allocation issues in a DS-CDMA cellular system using genetic algorithms (GAs) to obtain global optimal solutions, which are powerful and broadly applicable stochastic search and optimization techniques based on the principles from the evolution theory[2][3][4]. Most of the previous studies[5][6] assumed that the transmitter

^{*} This work is supported in part by the Grants-in-Aid for Science Research No.12680432 and by Research for the Future Program at Japan Society for the Promotion of Science

power level is controlled in a constant domain under the assumption of uniform distribution of users in the coverage area or in a continuous domain. In this paper, the optimal Centralized Power Control (CPC) scheme which helps in the design of distributed power control schemes that are easy to implement is characterized and its optimal solution for CPC is presented using GAs in a typical case[1] and a large scale of DS-CDMA cellular system under the realistic test that means random allocation of the active users in the entire coverage area. In other aspects, we tighten the balance of the signal-to-interference ratios at the receiver and the convergence rate [3][4][5] by GAs, that is used to measure the rapidity of capturing the optimal solution.

2 Transmitter Power Allocation Problem

In the cellular system, we assume N users and M base stations. All users use the common radio channel in the DS-CDMA system. Let p_i denote the transmitter power of user i so that $\mathbf{P}=[p_1, p_2, \dots, p_N]$ denotes the transmitter power vector of the DS-CDMA cellular system. The corresponding received signal power of user i at base station k is $L(i, k)p_i$ where $L(i, k)$ denotes the gain for user i to base station k . The interference seen by user i at base station k is $\sum_{j=1, j \neq i}^N L(j, k)p_j$. It is assumed that the system is interference limited and therefore noise is ignored. A mobile user i uses the base station k which is closest to it for communication. All the $L(j, k)$ s are greater than zero. The SIR of mobile user i at its base station k is then written by[7][8]

$$\begin{aligned} SIR_i &= \frac{p_i L(i, k)}{\alpha \sum_{j=1, j \neq i}^N p_j L(j, k)} \\ &= \frac{p_i}{\alpha \sum_{j=1, j \neq i}^N p_j G_{j,k}} \quad \text{for } 1 \leq i \leq N \end{aligned} \quad (1)$$

where

$$G_{j,k} = \begin{cases} \frac{L(j,k)}{L(i,k)} & \text{for } j \neq i \\ 0 & \text{for } j = i \end{cases} \quad (2)$$

and α is the voice activity factor. In order to achieve the balance of SIR, the optimized issue of the same SIR for all the users in the system is expressed as

$$\begin{aligned} SIR_{opt}^- &= \max_{P \geq 0} \min_{1 \leq i \leq N} SIR_i \\ SIR_{opt}^+ &= \min_{P \geq 0} \max_{1 \leq i \leq N} SIR_i \end{aligned} \quad (3)$$

where, SIR_{opt}^- and SIR_{opt}^+ are the maximum of minimum value of SIR_i and minimum of maximum value of $SIR_i, i = 1, \dots, N$, respectively. Due to the theorem and lemma of R.Vijayan and J.Zender[1], let us define G as an $N \times N$ matrix that has $G_{j,k}$ as its elements. The matrix G has a few important properties which are described as follows.

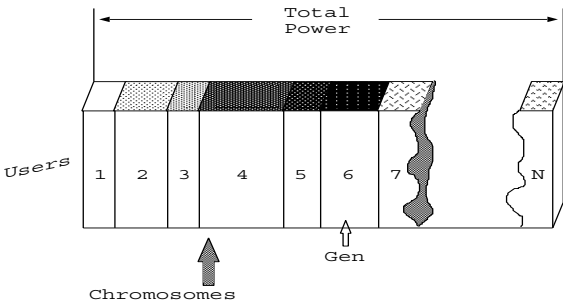


Fig. 1. Power allocation issue in CDMA system

- 1. G is an irreducible nonnegative matrix
- 2. There exists a unique SIR^* given by

$$SIR^* = SIR_{opt}^- = SIR_{opt}^+ \tag{4}$$

So we have the same SIR^* that is achievable by all users. In a large scale of DS-CDMA system with random allocation of users in its coverage area, it is not easy to find the optimized solutions. In this work, we adopt GAs to search them as fast as possible termed convergence rate for operators, helping in the implementation of distributed power control schemes.

The total power allocation used here consists of a string structure where the total length is the sum of the total power required by the total number of users in the system as shown in Fig.1. The components of the algorithm are examined in the following subsections.

2.1 Objective Function

The objective function will essentially determine the survival of each chromosome by providing a measure of its relative fitness. The primary goal of any approach to solve the power control problem is to decrease the multi-user interference, balance the services and achieve the optimized power allocation, while satisfying the required quality of signal transmission. By assigning the power to each user to satisfy the same SIR for all users, the objective function which encompasses all of the considerations is described as

$$\eta = \min_{P \geq 0} |SIR_{opt}^+ - SIR_{opt}^-| \tag{5}$$

2.2 Crossover

After reproduction, crossover proceeds with a probability, p_c . This operator takes two randomly chosen parent individuals as input and combines them to generate two children. This combination is achieved by choosing two crossing points in

the strings of the parent and then exchanging the allelic values between these two points. According to the fundamentals of CPC, preliminary simulation results show that with the simple crossover operator, a significant number of the configurations will be generated. In order to greatly speed up the convergence rate and the computation, the evolution is then proceeded via the partially matched (PMX)[2] crossover operator. At the same time, we introduce the discarding strategy of the invalid parents in the process of PMX crossover. On the other hand, the crossover points and number can be selected automatically according to the constraints of the power allocation. Furthermore, our solution representation allows us to further reduce the search space, it is very available for investigating the practical large scale of CDMA cellular wireless system and can search the optimized solution as soon as possible. We termed our crossover operator as adaptive partially matched crossover as APMX.

In order to achieve APMX easily, each individual is represented by the real number vector and use nonlinear arithmetic crossover to process them. We also create two First-In First-Out (FIFO) stacks to store the individuals. One is following the increasing step based on the amount of $SIR_i, i = 1...N$ of the individuals. Another one is following the decreasing step in the purpose of easy crossover and speeding up the convergence rate. When we select the parents, following the above principle of the two stacks. In this case, given two parents A and B , one is the individual with the maximum of SIR and the another one with the minimum of SIR . The crossover is performed by first generating cross points by adopting the gradually deceasing length of crossover during the entire crossover procedure. At the end of the GA's operators, we will rank the individuals by FIFO stacks and discard some individuals with the very significant changes between $SIR_{opt}^+ - SIR_{opt}^-$.

Because of a unique solution for our issue[1] and to speed up the convergence rate, we design the APMX algorithm in which nonlinear arithmetic functions are used in our crossover operator. The arithmetic crossover is defined as the combination of two chromosomes, $p_i(t)$ in t -th generation with the maximum SIR_{opt}^+ and $p_j(t)$ in t -th generation with the minimum SIR_{opt}^- as follows:

$$\begin{aligned} p_i(t+1) &= p_i(t) - \lambda p_j(t) \\ p_j(t+1) &= p_j(t) + \lambda p_i(t) \end{aligned} \tag{6}$$

where, for obtaining λ , the three types of nonlinear arithmetic exponent functions are introduced in the investigation as

$$\lambda = \frac{1}{\beta + \mu t} \tag{7}$$

where, β , and μ are control parameters , which will determine the convergence rate of the GAs.

2.3 Mutation

The mutation operator provides opportunities for long jumps from local minima because the crossover operator may lead to falling into a local minimum of the

fitness function as the generated children tend to be very similar to their parents. A low level of mutation serves to prevent any one element in the chromosome from remaining fixed to a single value in the entire population. On the other hand, a high level of mutation will essentially result in a random search. To maintain a balance between such extremes a valid value for p_m is 0.01[3][4].

2.4 Selection

The selection operator produces individuals with higher potential to be optimal solutions. The selection operator is very important as it must usually accomplish a trade-off between the two opposing and undesirable tendencies. Thus, individuals with higher fitness have more chances to reproduce. In the other hand, if only the fittest individuals are selected for generating new generation, it may result in a quick convergence rate to local optimal solutions. Therefore in FIFO stack, we adopted two stacks with stack depth, N . The same parents, A and B will be selected two chances for reproducing new children. From the two time number of children, it is more available for us to select fitter new generation and at first, discard some worst individuals. With this procedure, the fitter individuals have a higher probability of being chosen than weaker counterparts. According to the simple selection based on a roulette wheel, the probability of any individual to be selected from the population may be defined as

$$\psi(i) = 1 - \frac{|SIR_{opt}^+ - SIR_{opt}^-|}{\sum_{j=1}^N SIR_j} \tag{8}$$

2.5 Termination Criteria

In this paper, to achieve the balance of the signal-to-interference ratios at the receivers and speed up the convergence rate adopting GAs, in which the individual strings in the current population are processed by the genetic operators described above subsections to form new generation. Whether the best candidate in the generation does not violate any of the problem's constraints, the search may terminate. In each iteration step, the search can also be terminated when there are no significant changes in the difference between the two successive generations depicted as

$$\eta(t) = \min_{P \geq 0} |SIR_{opt}^+(t) - SIR_{opt}^-(t)| \tag{9}$$

Then, the search of GAs will be implemented under the following conditions satisfied as

$$\eta(t) \leq \delta \tag{10}$$

where δ is the termination parameter.

3 Simulation Results

3.1 A Typical Example

Following Ref.[1], the example was illustrated by CPC scheme in a smaller scale with only three mobiles using the same channel with link gains given by G matrix as follows:

$$\begin{pmatrix} 1.0 \times 10^{-4} & 4.82253 \times 10^{-9} & 3.57346 \times 10^{-10} \\ 1.52416 \times 10^{-8} & 6.25 \times 10^{-6} & 3.50128 \times 10^{-9} \\ 7.67336 \times 10^{-10} & 2.44141 \times 10^{-8} & 1.23457 \times 10^{-6} \end{pmatrix}$$

Power control simulation is done for each individual, which is assumed for its target as maximum as possible. Figs.2 and 3 show the results of the situations with and without FIFO. Genetic algorithm with FIFO and with a random initial value of power allocation to each user prove to be superior to the case without FIFO in 50 generations because in Fig.2 to obtain $\delta = 0.01$ over 200 generations will be necessary, on the other hand, the convergence rate will be in 160th generation with FIFO. We also see in Fig.2 it takes a longer execution time to obtain better results and Fig.3 makes more rapid progress in convergence rate than Fig.2. On the other aspect about the nonlinear arithmetic exponent function, when $\beta = 1$ and $\mu = 1$, the re-allocated power between two chromosomes will be largest which lead up to the best convergence rate.

For this typical example, if we use equal transmitter powers, the three users' SIR's are as 42.85 dB, 25.23 dB and 16.90 dB, respectively. After the simulation by GAs, we also obtain the three equal results of SIR's as 24.74 dB as the same results depicted in Ref.[1]. We also picks the unique solution of the power

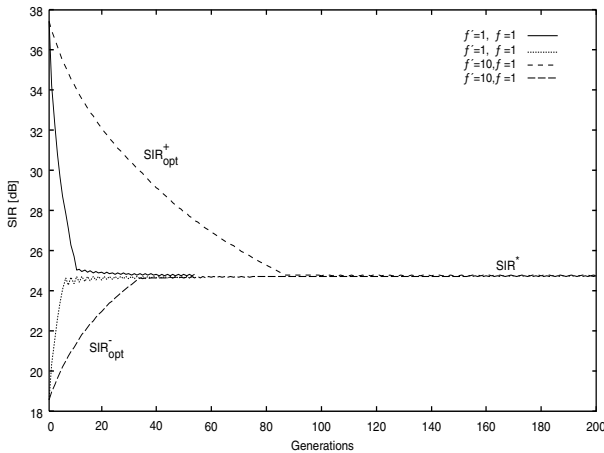


Fig. 2. SIR_{opt}^+ and SIR_{opt}^- versus the generation without FIFO ($\delta = 0.01dB$, $\alpha = 1$, $N = 3$)

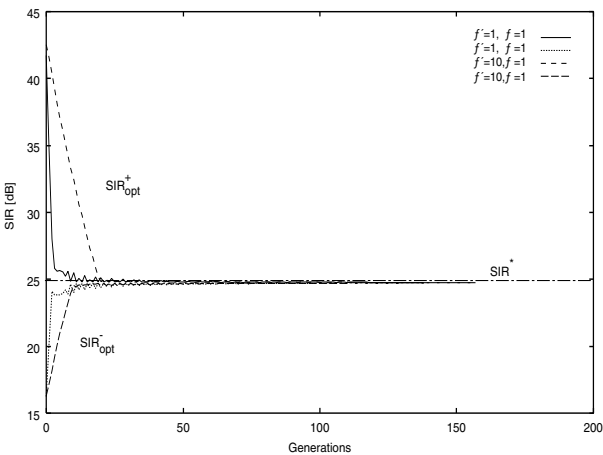


Fig. 3. SIR_{opt}^+ and SIR_{opt}^- versus the generation with FIFO ($\delta = 0.01dB$, $\alpha = 1$, $N = 3$)

allocation problem up as 1.79×10^{-3} , 8.67×10^{-2} and 5.11×10^{-1} , respectively. We see an improvement of 7.8 dB in minimum SIR by CPC strategy.

3.2 A Large Scale Cellular Wireless System with CDMA

In our simulation environment, we consider a general multi-cell CDMA cellular system on a rectangular grid. In this system, there are nine base stations with (x, y) coordinates $(1000i+1000, 1000j+1000)$ for $0 \leq i, 0 \leq j$. The x and y coordinates of each user are independent uniformly distributed random variables between 0-60km. Figure 4 shows the positions of base stations and the example of randomly distributed users in the system when we set $N_c = 30$ users/cell.

Figure 5 shows the users' SIR's versus the user number when we set the equal power to each user. We see the the rapid variations of each user's SIR are occurred that means some users have the better transmitting quality, and some have worse quality, which could not satisfy our purpose for balancing the services, especially in an integrated wireless cellular system.

We see the GAs with FIFO has the better astringency to obtain the unique optimal solution. In the investigation of the large scale cellular system, if the FIFO strategy is not adopted, it takes a much longer execution time for CPU. For the real time problem, this strategy will lose its purpose of solving CPC problem in a large scale cellular system. On the other hand, it might lead up to no astringency by GAs because these methods are blind search techniques for finding optimal solutions in the entire solution space. Then we use the FIFO strategy in the investigation of the large scale cellular system. Figure 6 shows the convergence rate of the users' SIR's with maximum value of SIR and with minimum value of SIR. We see that SIR^* reaches the target optimal value in

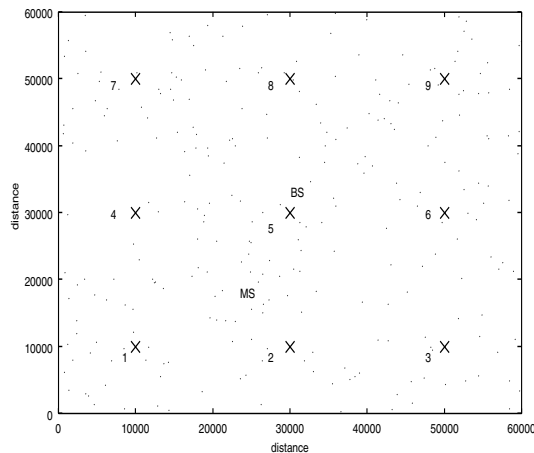


Fig. 4. Simulation environment for the number of active users, N_c and nine cellular wireless system ($N_c = 30$ users/cell)

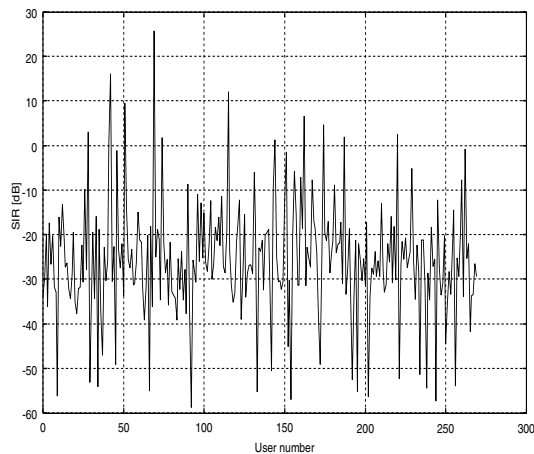


Fig. 5. SIR [dB] versus the user number without power allocation ($p_1 = p_2 = \dots p_N = 1$, $\alpha = 0.375$, $N = 270$)

over 50 generations as $\beta = 1$ and $\mu = 1$. As β increases to $\beta = 10$, it takes the better results as soon as possible near 15 generations. It performs much better in this simulation runs.

According to the simulation results, the final unique optimal solution, that is the best SIR^* , takes as -11.812542 dB whatever the nonlinear arithmetic exponent functions are used in GAs. In order to achieve this purpose, the power allocation graph by CPC has been obtained and shown in Fig.7 for the system

located in 3th and 6th cells slowly varies and with a little amount power required because of with smaller user density in this area shown in Fig.4.

4 Conclusions

According to the simulation results, it showed that genetic algorithms are robust for the optimal power allocation.

In this investigation, to speed up the convergence rate and filter out the illegal solutions, we introduced the nonlinear arithmetic exponent function and FIFO strategy which may improve the convergence rate of genetic algorithms. Then we effectively simulated the centralized power control in a real, large scale cellular wireless system and obtained the better results.

The main benefit of these simulation results is that they provide an estimate of CPC and as the basics for the design of DPC in the system. Furthermore, they provide the reference results when we design the burst admission algorithms or the system with varying processing gains.

References

- [1] S.A.Grandhi, R.Vijayan, D.J.Goodman and J.Zender, "Centralized Power Control in Cellular Radio System," IEEE Trans. on Veh. Technol., Vol.42, No.4, pp.466–469, Nov. 1993
- [2] W.K.Lai and G.G.Goghill, "Channel Assignment Through Evolutional Optimization," IEEE Trans. on Veh. Technol., Vol.45, No.1, pp.91–96, Feb. 1996
- [3] T.P.Wang, S.Y.Hwang and C.C.Tseng, "Registration Area Planning for PCS Networks Using Genetic Algorithms," IEEE Trans. on Veh. Technol., Vol.47, No.3, pp.987–995, Aug. 1998
- [4] C.Y.Ngo and V.O.K.Li, "Fixed Channel Assignment in Cellular Radio Networks Using a Modified Genetic Algorithm," IEEE Trans. on Veh. Technol., Vol.47, No.1, pp.164–172, Feb. 1998
- [5] R.Prasad, *CDMA for Wireless Personal Communication*, Boston, MA: Artech, 1995
- [6] V.K.Garg and J.E.Wilkes, *Wireless and Personal Communications System*, IEEE Press, 445 Hoes Lane, Piscataway, NJ, 1996.
- [7] F.Adachi, "Theoretical Analysis of DS-CDMA Reverse Link Capacity with SIR-Based Transmit Power Control," IEICE Trans. on Fundamentals, Vol.E79-A, No.12, pp.2028–2035, Dec. 1996
- [8] R.D.Yates and C.Y.Huang, "Integrated Power Control and Base Station Assignment," IEEE Trans. on Veh. Technol., Vol.44, No.3, pp.638–644, Aug. 1995

The Coexistence of Multicast and Unicast over a GPS Capable Network

T. Asfour and A. Serhrouchni

Ecole Nationale Supérieure des Télécommunications (ENST)
46 rue Barrault, 75634 Paris Cedex 13, France
{asfour, ahmed}@enst.fr

Abstract. The last few years have seen a great number of new network applications. Many of these applications are characterized by their multipoint features and by their needs of high bandwidth. To allow the deployment of these applications at a large scale in the Internet without running the risk of network implosion, an adapted congestion control mechanism has to be used. In this paper we propose to use the *Fair Scheduler paradigm* for end-to-end congestion control for the design of a congestion control algorithm to be used with the Reliable Multicast Transport Protocol RMTP. The FS paradigm supposes that all the network routers use a fair scheduler queueing mechanism. This hypotheses allowed us to have worst case bounds on the queueing delay that a block of RMTP data packets experiences between the sender and any receiver in the multicast group. Based on these bounds the sender will regulate its transmission rate in order to avoid a network congestion. Two different versions of this algorithm are presented in detail. The coexistence RMTP/TCP will be the issue of a suite of tests that intend to evaluate the performance of the new algorithm over a Fair Queueing (FQ) scheduled network with the presence of TCP flows sharing the same network resource as the RMTP flow. The same tests will be rerun over a FIFO scheduled network in order to have an idea about the advantages of using the scheduling discipline as a network support for congestion control in the Internet.

1 Motivations and Related Work

The permanent upcoming of new applications that do not use the unicast Transport Control Protocol TCP [10, 11, 2] as the underlying transport protocol risks to destabilize the TCP/IP Internet. These new applications are characterized by their multi-point features like audio/video-conferencing, multi-party games and file distribution. Faced with the fact that TCP is not appropriate for multi-point transport and that multicast applications have different needs a general consensus in the Internet community seems to be that there isn't a single multicast transport protocol able to satisfy the needs of all multicast applications. In consequence a lot of multicast transport protocols are specified, some are for real time applications like RTP/RTCP [8, 7], others for bulk data transfer like

MFTP [9], etc. Besides the big number of Internet users (300 million users are expected in 2001) the upcoming of these new multicast applications increases Internet traffic exponentially. To prevent the explosion of the Internet it becomes imperative to determine some rules of coexistence between unicast and multicast protocols. This paper addresses mainly this issue. We will focus on the transport layer protocols where a real conflict arises between these protocols due to the Internet resources limitation and the absence of deterministic rules that allow unicast and multicast control functions to share these resources in a fair manner. In the current Internet these control functions are located at the end hosts, the network is still passive and does not play an active role in the control of data flows. The TCP-friendly paradigm [3] was introduced to devise congestion control protocols compatible to TCP. However it seems to be unable to satisfy the needs of new upcoming applications, like multipoint applications. A TCP-friendly source-based congestion control scheme for multicast flows has to adapt its sending rate to the worst receiver (in terms of loss). Numerous multicast flows would suffer from the application of this scheme, including RMTP. Therefore, new research efforts are oriented toward new schemes for end-to-end congestion control. These schemes intend to satisfy the needs of more applications without destabilizing the Internet. The Fair Scheduler (FS) paradigm [4] is one of these schemes.

In a previous publication [1] we have presented a simplified model of RMTP (Reliable Multicast Transport Protocol) that we have simulated using the Network Simulator NS. In our model RMTP uses a TCP-like end-to-end congestion control mechanism. Based on this model we have tested the performance of the protocol over a FIFO scheduled network. The simulation results show the bad performance of the protocol when used to multicast data to receivers with very heterogeneous criteria like bandwidth, loss rate and delay, and that it is up to the worst subgroup in terms of these criteria to determine the performance of the protocol. The simulation shows as well the impact of what we have called the “loss misunderstanding” problem on the performance of the protocol. This problem appears when the sender considers packets as lost that in reality are still in transit to the receivers and not yet acknowledged when the new sending interval starts, which prevents the sender from advancing its window. This phenomenon is mainly due to the fact that over a FIFO scheduler network it is not possible to have worst case bounds on delay. This makes the adjustment of the transmission rate in function of the group’s physical conditions very difficult.

The organization of this paper is as follows: In Section 2 we present briefly the FS paradigm and we propose a new congestion control algorithm for RMTP based on this paradigm. The performance evaluation of the new RMTP congestion control algorithm will be presented in details in Section 3, and finally Section 4 concludes this paper.

2 FS Based Congestion Control Algorithm for RMTP

In general, support from the network can help congestion control. However this support, that can go from simple buffer management to active networking, is still not completely trusted by the Internet community. This distrust is mainly due to the aim of not violating the end-to-end principle of the Internet and to the lack of a clear idea about what kind of router support to use. One of the simplest way to use network support is to change the scheduling discipline inside the routers. In [4], Biersack and Legout propose to deploy the PGPS-like¹ scheduling inside routers in a new congestion control paradigm called FS paradigm. The main assumptions of the FS paradigm are: A fair scheduler network, *i. e.* a network where every router implements a fair scheduler, with end users that are assumed to be selfish and non collaborative. A user acts selfish if he only tries to maximize his own satisfaction without taking into account the other users. When users do not use the same congestion control mechanism, they are considered as non-cooperative. In this section we will design a congestion control algorithm for RMTP based on the FS paradigm.

2.1 Work Context

We consider an RMTP virtual tree, as shown in Figure 1. Receivers are grouped into subtrees, called *local groups*. For each subtree, one receiver is chosen to be a local group master (called *designated receiver* or *DR*). Each DR is responsible for processing receivers' status messages and for achieving local recovery in its local subtree. The sender sends data via multicast to all the receivers in the group. Receivers in turn send their status messages periodically to the associated DR. Each DR plays the role of the sender in its subtree and retransmits lost data in its subtree. The DRs send their status messages periodically up to the sender, which in turn processes these messages and performs global recovery. Our RMTP model uses a hybrid window-based and rate-based congestion control mechanism. The sender sends data via multicast in regular intervals with a period of *interval*_. The maximum number of data that can be sent in one interval is $Ws\_$. So during one interval the maximum transmission rate is $Max\_Tr = Ws\_/interval\_$.

2.2 Worst Case Bounds on Delay for RMTP Session

To determine the value of Max_Tr we will proceed as follows: For a given value of $Ws\_$ we will compute, thanks to the FS paradigm's first assumption, the value of $D_{max}(1, k)$ that represents the maximum queueing delay the $Ws\_$ packets experience on a network path between the node 1 and the node K . In general a GPS² server m that serves N sessions on a link is characterized by N positive real numbers, $\phi_1^m, \phi_2^m, \dots, \phi_N^m$. These numbers denote the relative amount of

¹ Packet General Processor Sharing [13, 12, 5, 6]

² General Processor Sharing

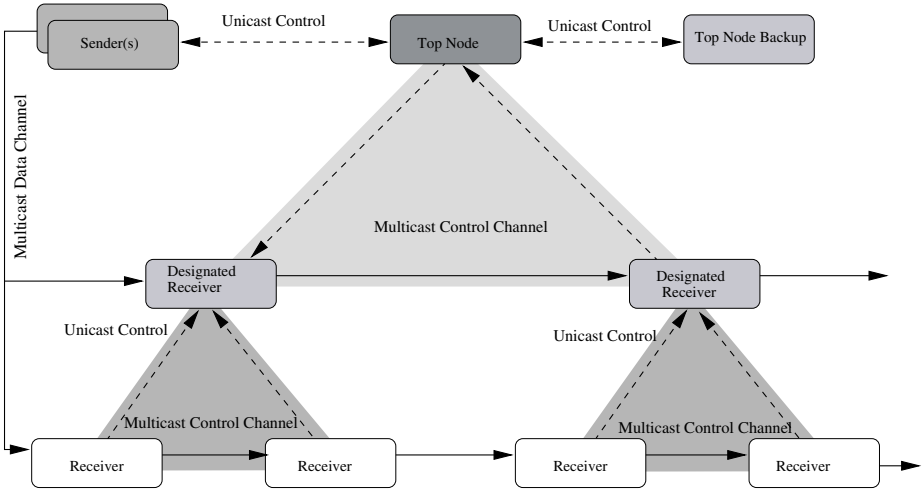


Fig. 1. RMTP Architecture

service to each session in the sense that if $S_i^m(\tau, t)$ is defined as the amount of session i traffic served by server m during an interval $[\tau, t]$, then

$$\frac{S_i^m(\tau, t)}{S_j^m(\tau, t)} \geq \frac{\phi_i^m}{\phi_j^m}, j = 1, 2, \dots, N \quad (1)$$

$$g_i^m = \frac{\phi_i^m}{\sum_{j=1}^N \phi_j^m} r^m \quad (2)$$

for any session i that is continuously backlogged in the interval $[\tau, t]$. (A session is backlogged at time t if a positive amount of that session's traffic is queued at time t .) Thus 1 is satisfied with equality for two sessions i and j that are both backlogged during the interval $[\tau, t]$.

From Equation 1, whenever a session i is backlogged it is guaranteed a minimum service rate of g_i^m , where r^m is the rate of the link represented by node m . g_i^m is called the session i backlog clearing rate, since a session i backlog of size q is served in at most $\frac{q}{g_i^m}$ time units. In consequence $D_{max}(1, k)$ is computed using Equation 3.

$$\begin{aligned} D_{max}(1, k) &= \sum_{n=1}^{k-1} \frac{Ws \cdot packetsize}{g_{rmtp}^n} \\ &= \sum_{n=1}^{k-1} \frac{Ws \cdot packetsize}{\frac{\phi_i^n}{\sum_{j=1}^N \phi_j^n} r^n} \end{aligned} \quad (3)$$

The propagation delay $D_p(i, j)$ between two consecutive nodes i and j on the path from node 1 to node K is considered to be constant. So the overall

propagation delay from node 1 to node K is $D_{prop}(1, k) = \sum_{n=1}^{k-1} D_p(n, n+1)$ and the maximum overall delay is $D = D_{prop}(1, k) + D_{max}(1, k)$.

2.3 Optimistic RMTP Congestion Control Algorithm

In this case the DR will acknowledge positively the packets it receives during an interval without waiting for the reception of the receivers' status messages. In other words we consider that if a packet is well received by the DR then the DR will be able to retransmit this packet to receivers in its local group. However, to assume this responsibility the DR has to keep a copy of every received packet in its local buffer until this packet is acknowledged by all the receivers in the local group. The sending interval is considered to be composed of two consecutive phases: a sending phase and a listening phase with periods $T1$ and $T2$ respectively. $T1$ is equal maximum delay requested to receive the $Ws\_$ packets by any of the DRs, *i. e.* $T1 = D_{total}(Sender, Dr)$, while $T2$ represents the time the sender spent listening to the status messages of the DRs.

When using the optimistic algorithm the sending interval has to be long enough in order to allow the DR to retransmit lost packets in its local group before the next interval starts. In general it might be necessary that a DR makes more than one retransmission cycle in order to assume a fully reliable transfer to all receivers in its local group. If we consider that the number of these retransmission cycles is N , then the value of *interval_* can be computed using the following formula:

$$D_{total}(Dr, Receiver) + \sum_{i=1}^N [D_{hack}(Receiver, Dr) + D_{lost,i}(Dr, Receiver)] \leq D_{hack}(Dr, Sender) + T2 \quad (4)$$

where $D_{total}(Dr, Receiver)$ is the maximum delay the $Ws\_$ packets experience between a DR and any of the receivers in the associated local group, $D_{hack}(Receiver, Dr)$ is the maximum delay a HACK experiences between any receiver and the associated DR, $D_{lost,1}(Dr, Receiver)$ is delay that is required to retransmit the lost packets sent by the sender, and $D_{lost,i}(Dr, Receiver)$ is the maximum delay required to retransmit the newly lost packets retransmitted by the DR in the $(i-1)^{th}$ retransmission loop. We define a "retransmission loop" as the associated events (*Receivers send HACKS, DR retransmits lost packets*). N is the number of retransmission loops required to have zero loss. In consequence the value of *interval_* has to satisfy the relation $interval\_ \geq D_{total}(Sender, Dr) + T2$. The pre-determination of the value of $T2$, and in consequence the value of *interval_*, will limit the choice of N to an upper bound. When the loss rate increases, this upper bound might not be sufficient in order to have enough retransmission loops to achieve full reliability, and this is exactly the inconvenience of determining N in function of *interval_*. On the other hand if *interval_* is determined in function of the required N , the delay of the session will increase and the throughput will decrease. It is quite difficult to choose the optimal value of

the tuple $(interval\_ , N)$ since this will be extremely dependant of the loss rate that receivers suffer from.

2.4 Pessimistic RMTP Congestion Control Algorithm

In this case the DR has to wait for the acknowledgments of the receivers in its local group. Here it is very important that the DR performs at least one retransmission loop before it aggregates the status messages of receivers in its local group with its own status message and sends a single status message back to the sender. This is illustrated in the following relation:

$$D_{total}(Sender, Receiver) + 2 \cdot D_{hack}(Receiver, Dr) + D_{lost,1}(Dr, Receiver) + D_{hack}(DR, Sender) \leq interval\_ \quad (5)$$

3 Performance Evaluation of the New RMTP Congestion Control Algorithm

In this section we are going to evaluate the performance of the optimistic and the pessimistic congestion control algorithms presented previously in two cases: The first case is that of a single RMTP session with the absence of any other unicast or multicast flows on the network. The second case is that of a single RMTP multicast session that has to share the network resources with multiple TCP sessions. The network topology in the two cases will be one sender, one DR and a set of 10 receivers associated to the DR, as shown in Figure 2. The performance metrics are: The number of retransmissions carried out by the DR, the mean average delay per packet for all receivers and the average throughput of all receivers. In all the tests the value of $WS\_$ will be set to 20 pkt/s. The simulations have all been done using the network simulator NS. The simulation time of all the tests is set to 100s. A suite of tests that intends to evaluate the performance of an isolated RMTP flow over a *Fair Scheduler* network in function of the error rate attached to each receiver was carried firstly using the optimistic algorithm (for $N = 1$ and $N = 2$) and secondly using the pessimistic algorithm. The results of these tests are shown in Figures 3, 5, and 7.

Another suite of tests that intends to evaluate the performance of the RMTP flow in function of the number of TCP flows that shares the same *Fair Scheduler* network resources has been carried out. The first TCP flow is between the same node as the RMTP sender and the first receiver in the multicast group, and the second is between the same RMTP sender and the second receiver in the multicast group, etc. This suite of tests has been carried out using the optimistic algorithm ($N = 2$) as well as the pessimistic algorithm. The results of these tests are shown in Figures 4, 6, and 8.

The previous results show that the throughput of the RMTP flow is generally superior using the optimistic algorithm compared to the pessimistic one. In contrast the delay difference between the two algorithms is very small, which means that the DR fulfills its role in local recovery efficiently and the need of global

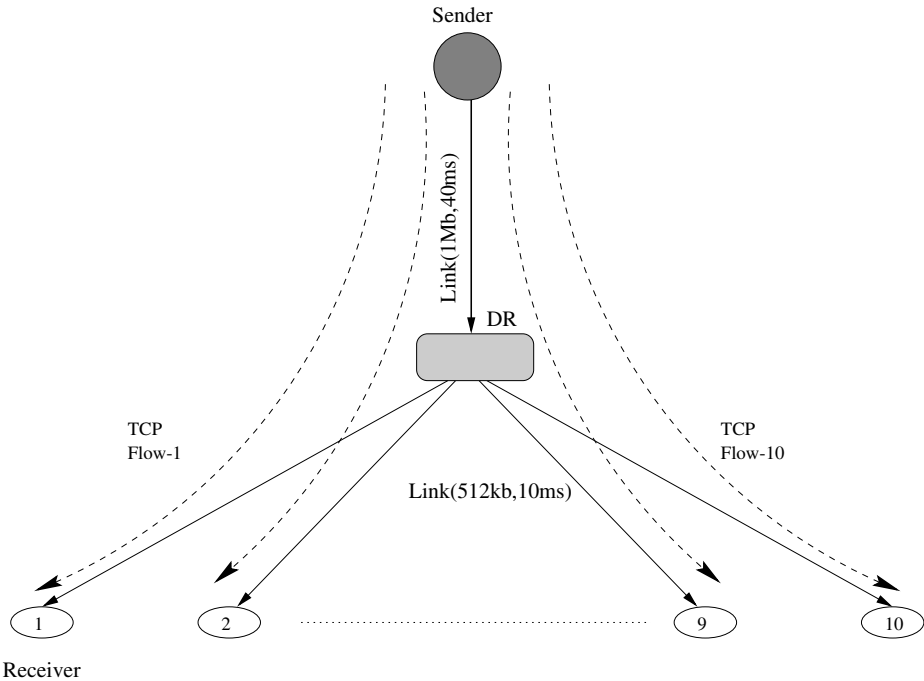


Fig. 2. Network topology

recovery in the case of the pessimistic algorithm is small. The simulation results show that the throughput of the RMTP flow decreases progressively with the number of TCP flows that share the same network resources. This was expected since the service the RMTP flow receives in each node of the network decreases when the node serves multiple flow in a fair manner.

To compare the performance of RMTP over a Fair Scheduler network with that over a FIFO scheduler network we have repeated the previous tests using the pessimistic algorithm over a FIFO scheduler network. The results of these tests are shown in Figures 3, 5, 7, 4, 6, and 8. The title of each of the corresponding graphs starts with “FIFO”.

The previous graphs show clearly that when the RMTP flow is the only flow in the network, the use of FIFO routers or FQ (Fair Queueing) routers give practically the same results. In contrast, when the RMTP flow enters in concurrence to other flows in order to share the limited network resources, the use of FQ routers gives better performance than that of FIFO routers.

4 Conclusions and Perspectives

In this paper we have presented our work concerning the design of a new end-to-end congestion control algorithm to be used with the Reliable Multicast Trans-

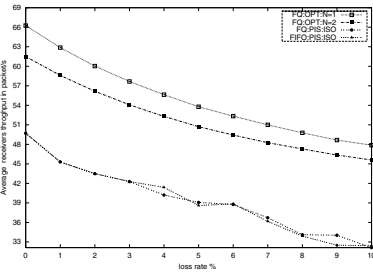


Fig. 3. Throughput results: Isolated RMTP

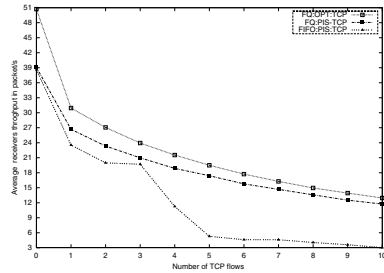


Fig. 4. Throughput results: RMTP with TCP

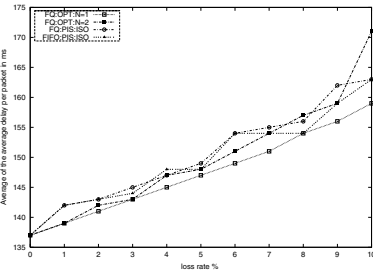


Fig. 5. Delay results: Isolated RMTP

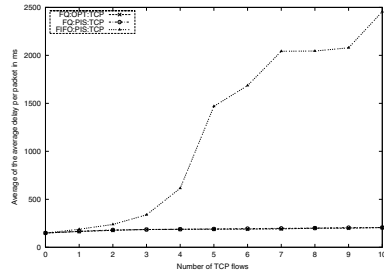


Fig. 6. Delay results: RMTP with TCP

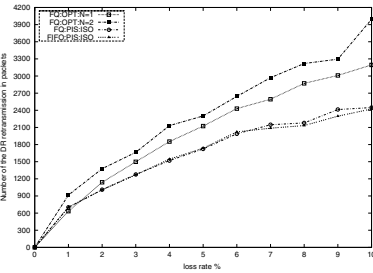


Fig. 7. DR retransmission results: Isolated RMTP

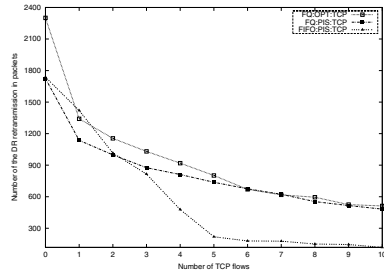


Fig. 8. DR retransmission results: RMTP with TCP

port Protocol RMTP. This algorithm is based on the FS paradigm for end-to-end congestion control proposed by Bierasck and Legout. The use of the FS paradigm allowed us to have worst case bounds on the queueing delay that a block of packets experience on a network path between two nodes. These worst case bounds are used to compute the sending interval of a RMTP source. In other words the use of the FS paradigm allows the RMTP sender to regulate its sending interval in function of the minimum service that the RMTP flow will receive in the nodes of the multicast tree. We have proposed two versions of this algorithm, an optimistic and a pessimistic version. In the optimistic case the DRs are considered to be able to retransmit packets in its local group under all circumstances. In contrast in the pessimistic algorithm the DR might not be able to achieve error recovery in its local group and therefore it will ask the sender for global recovery. The optimistic algorithm shows a superior performance compared to the pessimistic one in terms of throughput average.

The performance of RMTP with the presence of TCP flows on the network has then been evaluated in the case of a Fair Scheduler network and in the case of a FIFO Scheduler network. The simulation results in the first case show that the performance of RMTP decreases progressively when the number of TCP flows increases. In contrast, in a FIFO scheduler network, RMTP performance decreases significantly when the number of TCP flows increases. So when using the FS paradigm each multicast and unicast flow will receive always a minimum amount of service in all routers of the network, and their coexistence becomes more viable. In this paper we have focused on the performance evaluation of RMTP under the FS paradigm. However to complete this work we have to study the impact of this paradigm on the performance of TCP flows, which is one of the subjects of our future work.

References

1. Taghrid Asfour, Ahmed Serhrouchni, and Samir Tohme. Performance Evaluation of RMTP Using NS & RMTP Integration in current Networks. In *Proceedings of the INTERWORKING2000*, Norway, Bergen, October 2000.
2. Sidnie Feit. *TCP/IP: Architecture, Protocols, and Implementation*. McGraw Hill, second edition, 1996.
3. S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. Technical report, LBNL, May 1999. To appear in the IEEE/ACM Transactions on Networking, August 1999 .
4. A. Legout and E.W Biersack. Beyond TCP-Friendliness: A New Paradigm for End-to-End Congestion Control. Technical report, Institut EURECOM, Sophia Antipolis, France, nov 1999.
5. Abhay K. Parekh and Robert G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Transactions on Networking*, 1(3), June 1993.
6. Abhay K. Parekh and Robert G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple-Node Case. *IEEE/ACM Transactions on Networking*, 2(2), April 1994.

7. H. Schulzrinne. RTP Profile for Audio and Video Conferences with Minimal Control. RFC 1890, January 1996.
8. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889, January 1996.
9. StarBurst Software. StarBurst MFTP—An Efficient, Scalable Method for Distributing Information Using IP Multicast. <http://www.starburstcom.com/white.htm>, 1998.
10. W. Richard Stevens. *TCP/IP Illustrated: The Protocols*. Addison-Wesley, 1994.
11. W. Richard Stevens and Gary R. Wright. *TCP/IP Illustrated: The Implementation*. Addison-Wesley, 1995.
12. Opher Yaron and Moshe Sidi. Generalized Processor Sharing Networks with Exponentially Bounded Burstiness Arrivals.
13. Zhi-Li Zhang, Don Towsley, and Jim Kurose. Statistical Analysis of the Generalized Processor Sharing Scheduling Discipline. In *Proceedings of ACM SIGCOMM'94*, UK, London, August 1994.

An Approach for QoS Scheduling on the Application Level for Wireless Networks

Dang-Hai Hoang¹ and Dietrich Reschke

Technical University of Ilmenau,
D-98693 Ilmenau, Germany

{Hoang, Reschke}@prakinf.tu-ilmenau.de

Abstract. In this paper, we propose an architecture for Quality of Service (QoS) control with a proxy server for multimedia applications in heterogeneous communication environments of wired and wireless networks. The concept is based on a traffic scheduling mechanism on the application level for supporting various mobile user traffic streams. The proxy server is located in the base station and is responsible for user profile management and QoS adjustment. Via an application user interface, the user can click on a quality button to send an user feedback to express the actual required QoS. The concept is furthermore characterized by the gathering of queue lengths of packet flows and the calculating of the loss probability for QoS monitoring in the proxy server.

1 Introduction

Quality of Service (QoS) is an important factor in the service capability of the networks. QoS management is a topic of considerable amount of research over several years. The issue of QoS guarantees is no doubtable one of the great challenges in multimedia communication networks; not only in fixed networks including internet, but also in wireless networks.

Until now, most of the works have been in the developments of individual network components such as end-systems, transport and network protocols, protocols and functions in the management plane, etc. Some of works proposed general frameworks for QoS management, but most of them require changing of whole protocol structure. Summaries of QoS architectures can be found in [1][2]. Well-known are the QoS concepts related to layer three of the OSI reference model such as IntServ, DiffServ and above layer such as MPLS (Multiprotocol Label Switching). QoS concepts on lower layers are specially proposed for Local Area Networks (LANs) which are based on OSI layer two. Examples are Subnet Bandwidth Manager (SBM) for shared and switched 802 LANs such as Ethernet (also FDDI, Token Ring, etc.). Other layer 2 technology has been QoS enabled such as ATM (Asynchronous Transfer Mode). A lot of QoS architectures for end-systems were also proposed. Such architectures were shortly described in [1]. It was argued that there are commonalities exist between

¹ with a George-Forster Research Fellowship of Alexander-von-Humboldt Foundation.

QoS control and strategies found in end-system and network. This seems to be at the first glance, but there are different goals in designing control mechanisms between the intermediate system (routers, switches) and the end-system. The extent, to that the network-level QoS mechanisms can be applicable in the end-system (or vice versa) is still an open issue.

Naturally, the proposed concepts for fixed networks can be also applied with careful consideration for wireless networks. However, it is critical in wireless networks, where the wireless links are more error prone than the wired parts. On the other hand, there is a discrepancy in performance between desktop computers and handheld terminals of various types. The changes in network and technology have caused an increase of heterogeneities in distributed multimedia systems. Moreover, in mobile and wireless systems the quality of service of connections may change rapidly over time. Due to the varying transmission characteristics of the wireless links, the providing of QoS guarantee in wireless networks and heterogeneous environment is rather limited calling new directions of research in QoS treatment over wireless network environments. There is a need for mechanisms to compensate the performance gap, that means also the gap of quality of service which perpetually remains between different types of end-systems. The consideration of the user preferences will be also an indispensable necessity in future networks

In this paper, we present a framework for the design of such mechanisms which address the following three questions: where is the suitable location of QoS control? How can the gap of QoS be compensated? How can the QoS be monitored and adjusted? To answer these question, we propose a proxy server located in the base station. The proxy server receives the data stream sending from a fixed host, transforms and forwards the data packets to the mobile terminal according to the user profile and current available link bandwidth. The transmission of data packets occurs at the application level using a scheduling algorithm and a reference time slot system. The QoS is dynamically adjusted regarding the link bandwidth measured by the proxy server. At any time, the customer can send a user feedback to the proxy server for expressing his/her preferred QoS. We focus in this paper on QoS management of video streams, since it is the most critical component in the multimedia applications

The rest of paper is organized as follows. In section 2, we describe the background of this research and review related works. In Section 3, the proposed QoS management architecture is presented. Section 4 presents the experiment with the concept. Finally, Section 5 concludes this paper.

2 Background and Related Works

Many efforts have been done on constructing QoS architectures to support end-to-end QoS management. Most of these are predicated on the availability of guaranteed services and concentrated on the implementation of QoS mechanisms in networks and end-systems as well as their coordination in order to provide QoS support to the application [3]. However, multimedia applications have fluctuating needs on network resources which may also vary in heterogeneous environment. An overview of future

wireless internet access architectures is slightly given in [4]. The paper discussed the cornerstones of wireless internet access including proxy structures. Proxy structures enable the distribution of internet protocol operation between both the end-system and the network. Examples are WAP (Wireless Application Protocol), Snoop (a protocol booster) and ReSoA (Remote Socket Architecture). All these approaches take into account that the wireless links should be treated in a special manner.

In contrast to the main QoS issues in wired networks, the QoS degradation in wireless networks is not only due to the congestion, but also caused by possible high BER of the radio channels, fading and interference between channels etc. Thus, the usual retransmission for recovering lost packets is not suitable and not efficient any longer. The Forward Error Correction (FEC) has disadvantage of complex overheads. New solutions are necessary. To date, there are efforts made in improvements of radio link technologies, new and efficient compression techniques or improvements of the protocol stacks, etc.[5][6].

The future networks are likely to remain heterogeneous due to the increasing changes in network technology. Not a single technology can cover all aspects of communication applications. The coexist of various technologies and equipment devices restrain the overall QoS support for all user. A customer with desktop-type receiver certainly would accept other QoS perceptually than a customer with a handheld receiver. This gap of QoS should be compensated by a suitable QoS mechanism which should be flexible to adapt to wireless links and user preferences.

Recently, many QoS aware applications have been developed which are capable of adapting to their environment. Recalled, from the network' point of view, the QoS guarantee is the ability of the networks to fulfill the different request of the applications of customers. If the networks cannot satisfy the request of the users due to resources lacking, the applications are to adapt to this situation. Adaptive applications could adapt their operation to deal also with variations in QoS characteristics of networks. As an example is the play-out buffer adaptation [3]. The other possibility is the QoS on-demand approach [3][7][8] in which the user can actively react to achieve as a better QoS as possible. In this case, there is a need for controlling the user feedback and for monitoring QoS. Some QoS controlled applications were developed which enable the user to take control the QoS. In [8], Bechler et.al. distinguish three types of applications: common, adaptive and proactive applications. Unlike adaptive applications which are able to react to changes in the network condition, proactive applications actively affect the scheduling of resources. However, all these approaches either concentrated on the QoS manager in end-systems or did not consider the interactions between the applications and their behaviour.

In our approach, we suggest using a proxy server in the base station to compensate the gap between the different requirements of mobile terminals. Our approach differs from the proxy proposed in [7] in the fact that we suggest an active proxy structure. Active means that we use a traffic scheduling at the application level to manage the data streams sending to mobile terminals. Why is a traffic scheduling necessary? The reason is that we can give different priorities to the data streams sharing the same resource (buffer, bandwidth) in the base station. In case of lack of link bandwidth, it is necessary to decide how to drop data packets to ensure the QoS for certain

connections. In addition, we developed a graphical user interface that allows user to send the feedback signal to express the desired quality of connection.

3 QoS Management Concept on the Application Level

Figure 1 shows the principle of the proposed architecture in the context of wireless multimedia communication. The QoS proxy server takes the role of the QoS manager and the resource manager at the application level. In addition, it manages the user profiles. These components are described later in this section.

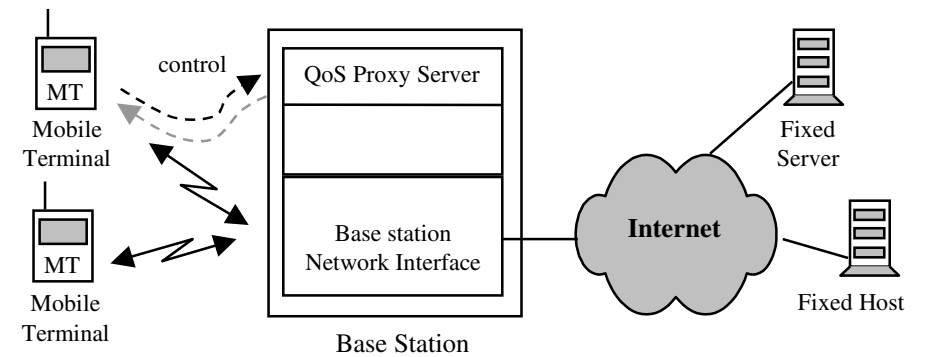


Fig. 1. QoS management architecture

As we discussed before, a mobile computer is more resource poor than a stationary computer and the properties of wireless networks are different from wired networks. Let us consider a situation where the fixed host sends too much multicast data to the mobile terminals when the link bandwidth of wireless network becomes low. In this case, the application typically should reduce the data rate sending to the mobile terminals, thus the QoS of all receivers will be affected. In a such a heterogeneous environment, the proxy server filters the data from the Internet and sends them to the mobile terminals according to their performance, QoS requirements and bandwidth allocation. We believe that this proxy structure is also advantageous for charging and billing purposes.

3.1 The Basic Concept

The proxy server consists of three functions: the QoS management, the resource management and user profile management. The proxy server receives and stores the user preference sending from the mobile terminal. Receiving data streams from the Internet, the QoS manager filters packets, converts data and sends the transcoded data to the receiver. The filter function was already mentioned in [7] and some other works (QoS-A pointed in [1]). If the bandwidth of the wireless link becomes low, the

QoS manager adapts the data streams to the actual bandwidth condition (e.g. discards unimportant packets, converts video to monochrome, etc.). The resource manager is responsible for monitoring the actual link bandwidth and allocating bandwidth to different data streams using a scheduling algorithm. The connections are given different weights according to the QoS requirements. For scheduling a queue system is used. A reference time slot system helps to define the priorities of the connections. At the application-level, it is advantageous for the transcoding function. In addition, the concept may not depend on the underlying network system.

A simple concept model is shown in figure 2 for a video transmission application. The receiver is a mobile terminal and the sender may be an another mobile terminal or a host in the wired network. For a video call, the call processing is as follows. The sender sends a connection setup message to the call manager of the wireless network. We suppose the call signaling was done by the wireless network and a connection is setting up. Now, the receiver has to send a short message containing his preference and his user profile is stored in the proxy server. This information are codec, frame rate, display size, color depth. These are the application-level QoS parameters. The proxy server set up a queue for the connection with a corresponding weight, i.e. the connection is allocated a bandwidth regarding to the weight. The sender application then admits a video stream to the respective queue in the base station. This data stream is then transcoded according to the user profile, i.e. the required QoS of the receiver and is sent to the receiver terminal. Pushing data packets into queues, the proxy server marks the data packets with respect to important and unimportant video frames. For determining the service order of the data packets, a bandwidth scheduler is proposed. We suggest to use an adaptive bandwidth scheduler proposed in [9], which is able to adjust the weights of connections. A reference time slot system is proposed regarding the actual order of packets that will be transmitted on the physical link.

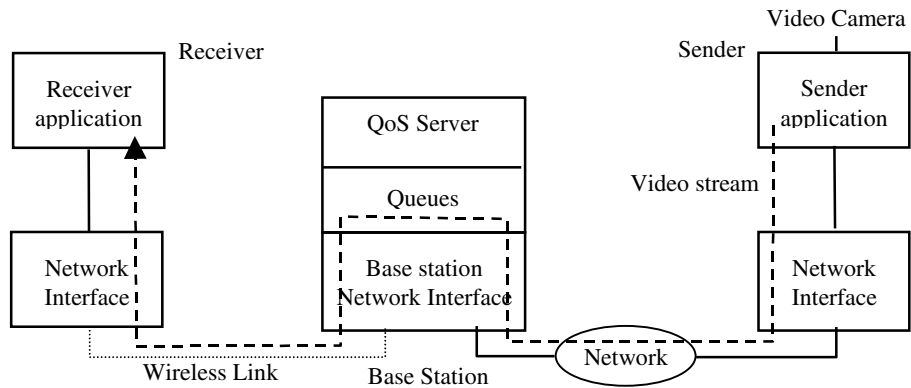


Fig. 2. The concept model for video transmission application

During the connection, the proxy server monitors the actual link bandwidth condition and adjusts QoS. The bandwidth is measured by the received data packets per time. The receiver has then to count the packets and sends it to the resource

manager. If the link bandwidth is very scarce, or when the receiver is not reachable, the proxy server discards unimportant frames or even all video frames and sends only audio during the critical time.

By gathering the queue lengths, a predication of QoS can be made. An increase of the queue length means the connection is worse treated or much data packets sending from the sender. In any case, there exists the risk of packet loss accompanying a QoS reduction. Two decisions can be made. Either the customer sends an user feed back to request a better QoS, or the proxy server adjusts the QoS itself with current bandwidth condition. An user interface is developed which allows the user to express his preference, i.e. the quality he/she wants to pay for, by clicking on a button on the display window. Upon his clicking, a user feedback signal is sent to the proxy server. The proxy server then varies the weight of his connection queue, thus he can receive more bandwidth in compare to other connections.

In following, we describe the components developed in the concept including the graphical user interface, the user profile, the user feedback and the QoS scheduler. The concept demonstration for video transmission was written in C and TCL/TK.

3.2. User Interface

The simple user interface is depicted in figure 3, consisting of a video display area and three main buttons. By clicking on the call button, the label is then changed to stop. The destination address is need to enter, either the IP address or the name of the target computer. The other button is designed for setting user preference. The user preference is selected at the call setup, it is also possible to send the preference to the proxy server during the call. The Quality button (minus or plus, i.e. worse or better QoS) is for user feedback.

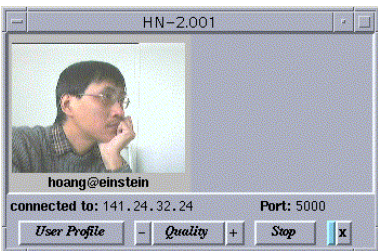


Fig. 3. Graphical User Interface

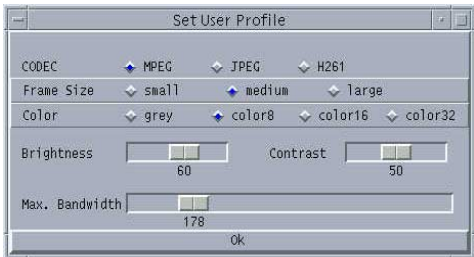


Fig. 4. Window for set user profile

The other button are for setting the video device such as port of the video card or setting the X-Window for test without a video camera. On the other side, the application receives the transcoded video stream from the base station, decodes it and plays out the video stream on the display area.

3.3. User Profile

The user profile contains information about the technical characteristics of user's mobile terminal. This information includes the display size, the color depth, etc. Furthermore, it includes the receiving preference of the customer, for example the desired codec, image size, brightness, contrast and the maximum bandwidth.

Clicking on the user profile button, the customer then set the desired parameter expressing the preferred application-level QoS. These parameters are then sent to the proxy server and stored in a user profile. Figure 4 shows the current defined parameters. Up to now, there have been various video data compression techniques defined by the standard organization. Thereunder are MPEG-1, MPEG-2 of Motion Picture Experts Group, H.261, H.263 of ITU (International Telecommunication Union). Other technique for still image compression is JPEG (Joint Photographic Experts Group). However, we use in the concept demonstration only three examples to consider the possibility of the proposed concept.

Parameters affecting the video playout are brightness, contrast, frame size and color depth. The user can specify these parameters using a simply click on the setting window or the slider bar. The frame size or resolution corresponds to the number of pixels in one frame. We assume to set one of three values: small (160x120), medium (320x240) and large (640x320). The brightness and contrast are preset to 60 and can be varied from 0 to 100. These parameters can be observed on the display of receiving window together with other information such as actual frame rate, actual received bandwidth, loss rate, etc. The maximum frame rate is 30, what is a typical limit for a video transmission.

3.4. User Feedback

The user feedback is performed via a simple interface with two buttons: plus and minus button on quality. By clicking on one of these buttons, the user sends a special message indicating `USER_FEEDBACK` to the proxy server. This express the desired quality of service of the customer. One way to do this is to send a single parameter indicating the desired quality, for example an utility value. The proxy server has then to map this value to the network parameter such as bandwidth, delay, loss etc. An algorithm is necessary to translate the application-level QoS parameters into the network QoS parameters. The mapping can be done by a translation function or a table.

However, we did not just developed the mapping function in the system. Instead of that, we use currently a simple way, in which we suggest that, the user will send the feedback message with an explicit wish on quality, e.g. bandwidth, image size, etc. We are investing on the development of a quality function to generally express the quality desired by the user. The quality function should express the utility of the user and is a function of the bandwidth, the efficiency of the transmission protocol and the error rate. This is the topic of next works.

3.5. QoS Scheduling and Monitoring

As mentioned above, the proxy server has three functions. The functions of QoS management and resource management are related closely together and form together with the queue system a QoS scheduling system. According to QoS requirements of the user, the proxy server set up the weight for the customer's connection.

The proxy server receives periodically the information about the link bandwidth measured by the receiver and adjusts the quality of connection to the actual link condition. In case of lack of bandwidth, the proxy server will decrease the frame rate, discard unimportant frames, decrease the color depth, or even discard all video frames, thus only voice is transmitted in this worse case. For this purpose, the video packets are marked as important and unimportant packets with an indicator in the packet header. The information about actual bandwidth is calculated based on the actual received packets in the receiver and is sent to the proxy server using a special message indicating BANDWIDTH_INFO.

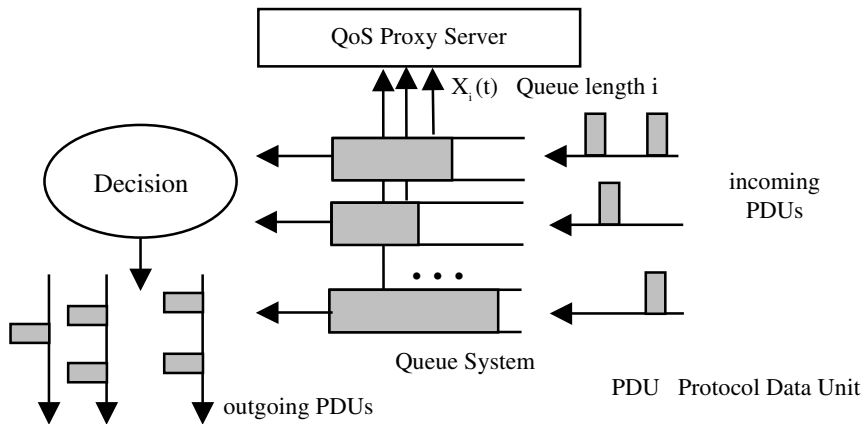


Fig. 5. The concept model for video transmission application

Figure 5 shows the principle model for QoS scheduling in the proxy server. As shown in the figure, the proxy server manages a queue system. Each queue is for one connection. Incoming packets from different flows arrive in the queue system. The queue lengths are gathered and the packet loss probability is calculated. Base on this probability and the feedback from the receiver, the proxy server monitors the QoS of packet flows and adjusts QoS to the actual link condition and the QoS requirements from the user. A separate queue is for each down link connection. We consider a reference time slot system according to 16 time slots of a W-CDMA frame (as described in the UMTS standards). In the packet mode, the base station has to select a packet from a connection during each down link time slot. By giving different weights on queues, we can determine the order of packet from different connections. Upon user feedback, the weight of the desired connection is changed. The decision process has then to select the packets from queues according to their weights, thus a virtual time slot order is realized. The incoming packets are marked according to

important or unimportant frames, as we mentioned above. For each queue, the scheduler maintains two parameters, namely the start tag ST and the finishing tag FT . The scheduling algorithm is defined as follows [9]:

$$ST_i = \max(TA_i^k, FT_{i-})$$

$$FT_{i+} = ST_i + \frac{L_k}{g_i(t)}$$

where i is the index of the queue i , TA is the arrival time of the packet k in the queue i , $g(t)$ is the allocated bandwidth for the queue i (weight of the connection), L_k is the packet size in bytes (including packet header, in fact it is the PDU size).

Whenever a packet arrives into a queue, the start tag and finishing tag are calculated in the scheduler. The decision is made based on the finishing tags, i.e. the packet with smallest finishing tag will be pushed at first to the layer below. By this way, the order of packets is determined. If the packet is first pushed into the lower layer, intuitively it will be also first served by the lower layer and will have preference before other packets by transmission into the wireless link. Without this virtual reference time slot system, the packets will be intuitive pushed into the below layer and served in the FIFO manner (First In First Out). Upon user feedback signal, $g(t)$ is adjusted and the finishing tag of corresponding queue is updated.

The relation between the bandwidth allocation and the queue length can be described by the following equation [9]: $C_i(t) = B * (1 + X_i(t))$, where B is a constant, X_i is the queue length of a queue i , C_i is the allocated bandwidth to the queue i . The probability of packet loss P is equal $P(X = X_m)$, where X_m is maximal queue length of a queue. That is the probability, by which the maximum queue length is exceeded. An increase of this probability means that the more packets in the queue. That is, either the sender is sending more packets or the link bandwidth becomes lower. It is the task of the proxy server to discard unimportant video packets or to adjust the weight of queue, i.e. to increase the bandwidth for the connection to keep up the desired quality.

4 Experiment Results

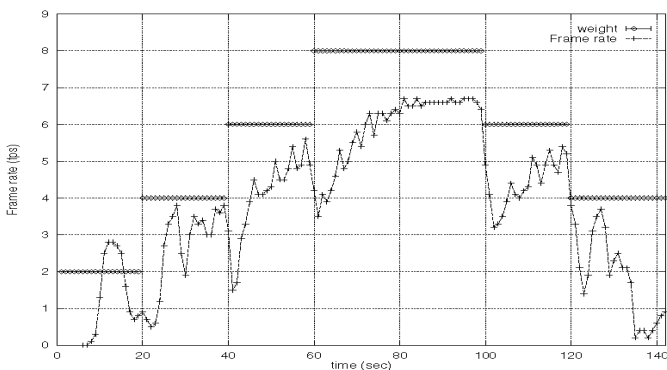


Fig. 6. Frame rate evolution regarding to weight

We have experimented using a laboratory testbed consisting of three computers: a video sender, a video receiver and the proxy server. Each computer is a Sun machine connected to each other over the Local Area Network. One key issue is the weight adjustment. The result of first experiment is shown in Fig. 6. The weight was increased by step of 2 at times $t=20s$, $t=40s$, $t=60s$ and decreased at $t=100s$, $t=120s$ according to the clicks on plus/minus QoS-buttons, respectively. The adjustment step is an implementation issue and depends on the available bandwidth. The frame rate, i.e. the QoS is better by increasing weight. At times of change, the frame rate is shortly low due to the updating in the system.

5. Conclusion

In the paper, we have discussed the issues of providing QoS in wireless communication networks. We proposed an architecture for Quality of Service (QoS) control with a proxy server for multimedia applications in heterogeneous environments of wired and wireless networks. The concept is characterized by a QoS scheduling mechanism on the application level. The proxy server is located in the base station and is responsible for user profile management and QoS adjustment. We developed a graphical user interface that allows user to send the feedback signal to express the desired quality of connection. The concept is furthermore characterized by the gathering of queue lengths of packet flows and the calculating of the loss probability for QoS monitoring. The QoS is dynamically adjusted corresponding the link bandwidth measured by the proxy server.

We intend to investigate in future works on the mapping of application-level QoS into network QoS using an utility function. Furthermore, we intend to extend the concept by integration of the utility function, the QoS-scheduling, the loss probability and the QoS adjustment.

References

1. C.Aurrecochea, A.T.Campbell, L.Hauw, "A Survey of QoS Architectures", Multimedia Systems, 6,98, 1998.
2. QoSForum.com, Stardust.com, White Paper, "QoS Protocols and Architectures", 8.-99.
3. V.Witana, M.Fry, M.Antoniades, "A Software Framework for Application Level QoS Management", IEEE , 3.1999.
4. A.Wolisz, C.Hoene, B.Rathke, M.Schlaeger, "Proxies, Active Networks, Re-Configurable Terminals: The Cornerstones of Future Wireless Internet ", IST Mobile Communications Summit, October, 2000, Ireland.
5. ETSI TR 122 960, V3.0.1(2000-01): "Universal Mobile Telecommunications System (UMTS); Mobile Multimedia Services including Mobile Intranet and Internet Services", (3G TR 22.960 version 3.0.1 Release 1999).
6. ETSI TS 123 107, V3.2.0 (2000-03): "Universal Mobile Telecommunications System (UMTS); QoS Concept and Architecture" (3G TS 23.107 version 3.2.0 Release 1999).

7. Yamazaki, J.Matsuda, "*Adaptive QoS Management for Multimedia Applications in Heterogeneous Environments: A Case Study with Video QoS Mediations*", IEICE Trans. Commun., Vol. E82-B. No.11, Nov. 1999.
8. M.Bechler, H.Ritter, J.H.Schiller, "*Quality of Service in Mobile and Wireless Networks: The Need for Proactive and Adaptive Applications*", 33rd Annual IEEE, HICSS, 2000.
9. D-H.Hoang, D.Reschke, "*An Adaptive Priority Approach to Bandwidth Scheduling in High Speed Networks to the Support of Multimedia Applications*", IFMIP 2000, June 2000, Hawaii.

A Host-Based Multicast (HBM) Solution for Group Communications

Vincent Roca and Ayman El-Sayed

INRIA Rhône-Alpes - projet Planète
{vincent.roca, ayman.elsayed}@inrialpes.fr
<http://www.inrialpes.fr/planete/>*

Abstract. In this paper we argue that overlay multicast is an important technology for applications requiring a group communication service. With this approach end-hosts (running the application), dedicated servers and/or border routers automatically self-organize into a distribution topology where data is disseminated. This topology can be composed of both unicast connections and native multicast islands (e.g. within each site). Therefore it offers a group communication service to all hosts, even those located in a site that does not have access to native multicast routing.

One of the issues raised is the set up of an efficient and robust overlay topology. In this paper we discuss several possible solutions. We show in particular the benefits of having a centralized approach, of using redundant links and updating the topology based on a host stability criteria.

Keywords: group communications, multicast routing, overlay topology, application-level multicast

1 Introduction

Group communication traditionally requires that each node at each site has access to a native multicast routing service. If intra-domain multicast (within a LAN or a site) is widely available, this is different for inter-domain multicast. Today many ISPs are still reluctant to provide a wide-area multicast routing service [5]: there are technical reasons (many aspects are still research topics), marketing reasons (e.g. which pricing model) and an “egg and chicken” problem.

At the early years of the MBone, the traditional solution was to *set up a tunnel* to a site connected to the MBone. Because of its limitations, it is now banned from new native PIM-SM/MSDP/MGBP deployments [1]. Another solution is to use a *reflector* [4]. A reflector is a host connected to the multicast

* This work has been done in the DSE (Distributed Systems Engineering) IST 1999-10302 European project. The consortium includes industrial companies involved in the space business: Alenia Spazio S.p.a., EADS Launch Vehicles, and IABG, as well as technology providers and research center: Silogic, Societa Italiana Avionica (SIA), University of Paris 6 (LIP6), CNRS-LAAS, and D3 Group. More information can be found in the DSE web site: <http://cec.to.alespazio.it/DSE/>

backbone and which creates point-to-point connections to all the remote hosts that do not enjoy inter-domain multicast routing. If this solution creates hot points within the network, on the other hand it is set up for a limited span of time – the session duration – and for a limited number of groups – those of the session – unlike tunnels.

Neither of these solutions is satisfying even if reflectors are frequently used. One of the goals of overlay multicast (also known as Host-Based Multicast, End System Multicast, or Application-Level Multicast) is to enable every host to participate in multicast sessions efficiently, no matter whether they have access to native multicast or not.

2 General Overlay Multicast Specificities

2.1 Specificities Compared to Traditional Multicast

The Overlay Multicast (OM) approach differs in many respects from traditional multicast routing:

- First of all a forwarding node in the overlay topology can be either *an end-host* (i.e. running the application), *a dedicated server* within the site, or *a border router*. On the opposite traditional multicast trees only include core routers.
- With an overlay, *the underlying physical topology is completely hidden*. A directed virtual graph is created between all the nodes. The virtual point-to-point links are assigned a weight corresponding to the one-way distance between nodes (several metrics are possible). Undirected graphs can also be used if the possibility of having asymmetric routes is overlooked.
- Another consequence is that an overlay topology, built on top of the existing infrastructure, can *integrate different flavors of multicast and unicast protocols* (e.g. several areas running different intra-domain protocols like PIM, MOSPF, DVMRP can be connected together in an OM topology).
- In traditional multicast, the membership knowledge is distributed in the core multicast routers. With an OM *group members are known* either by a Rendez-vous Point (RP)¹ [6], by the source, by everybody or is distributed among members [3].
- The *overlay topology is potentially completely under control*. For instance, our proposal takes advantage of the additional knowledge centralized at the RP (node identity, distances, node/link specificities) during the topology creation process.
- Yet a major drawback of involving end-hosts as transit nodes is that it *reduces the reliability* of the group communication service. Indeed an end-host is less reliable than a router or a physical link. For instance, if the overlay service is implemented as a library, then the node disappears if the application crashes or is stopped. Simulation results are given in section 3.

¹ This OM RP is different from PIM-SM RP.

- Another drawback is that the scalability of OM is lower than that of native multicast. This is another motivation for using multicast areas in the overlay whenever possible as all the nodes of this area are hidden behind a single OM correspondent.

3 Our Proposal: Host-Based Multicast (HBM)

3.1 Sketch of Our HBM Proposal

Our HBM proposal [10] distinguishes *core members* (CM) that are part of the core distribution topology and *non-core members* (nonCM) that graft on the existing topology as leaves. This distinction is based on several criteria explained in section 3.2. Everything is under the control of a central *rendez-vous point* (RP). This RP knows CMs and nonCMs and the distance between them (several metrics are possible here). This RP is responsible of calculating the OM topology and informing CMs/nonCMs. CMs periodically evaluate the distance between them and inform the RP. Likewise nonCMs evaluate their distance with CMs and inform the RP.

Of course this scheme:

- *has a limited scalability...*

More generally any OM solution based on point-to-point communications has scalability problems (even if having a central RP in HBM adds some more limitations). Yet many collaborative work sessions only include a limited number of hosts/sites and scalability is not a problem then. Besides a single HBM node can easily serve many local participants using the locally available multicast.

- *and greatly relies on the RP reliability...*

If the RP is collocated with the primary source (if any), then this is not an issue as any failure of the source host would anyway compromise the service.

On the other hand:

- *this is a simple solution...*

As all the information is centralized in the RP, there is no coherency problem and it does not create too much load on the nodes (an asset in case of lightweight hosts like PDAs). This is completely different in distributed solutions like [3] where each node runs various algorithms for group maintenance and incremental mesh quality improvement.

- *which can easily create a “not too bad” topology...*

The topology is optimal with respect to the distance database at the time of its creation². The update frequency of the distance database depends on various criteria like the group size (the larger the group, the lower the frequency) and node specificities (a powerful workstation can update the database more frequently than a PDA).

² More precisely, it is only limited by the ability of the topology solver to find an optimal solution.

3.2 Offering a Robust Group Communication Service

We argue that robustness is a key issue to OM solutions which are intrinsically fragile (sections 2.1 and 3). To improve it we introduce three mechanisms that all take advantage of the centralized knowledge at the RP.

The Need for Redundancy. First of all we add some redundancy in the topology. An algorithm, presented in Annex A, adds a certain number of Redundant Virtual Links (RVL) until the probability of having a partitioned topology after a node failure falls below a predefined threshold. This solution is not source dependent and therefore the OM robustness is the same no matter how many and where the sources are.

Of course some loops are created. Yet RVL are clearly identified as such and using a simple suppression mechanism is easy:

```

if (node N receives traffic both on the OM link and RVL)
    send a SUSPEND message on the RVL
    // on receiving a SUSPEND, the peer stops forwarding packets on the RVL
    // during a few seconds
if (node N receives traffic on the RVL but not on OM link)
    // there is a problem, yet N still receives new traffic and can
    // forward them on the OM
    wait some time and send a failure report to RP if situation persists

```

The Need for Fast Failure Discovery and Recovery. Robustness also requires that HBM node failures are rapidly discovered. This feature depends on the distribution topology in use (section 3.3):

- with a ring, one or two nodes in the ring must receive each packet twice, once in each direction. Otherwise there is a failure.
- with a shortest path tree, ACK messages can be generated by the leaves and aggregated by the transit nodes as they are sent back to the source. A transit node that does not receive an ACK from one of its downstream neighbors can easily conclude that there is a failure.
- using RVL provides a way to detect some failures rapidly. Yet multiple simultaneous failures may not be detected.

Each time a failure is detected, the topology is updated. Depending on the failure, this update can either completely reorganize the topology or just a subset of it (e.g. a partitioned area can be graft on the closest active transit node even if the new topology is sub-optimal).

Note that failures are usually due to application stop or crash, more rarely to link failures or WAN routing problems. Therefore a partition in the overlay topology does not prevent individual nodes and the RP to communicate using point-to-point connections.

The Need for Adaptation. Some of the nodes can turn out to be unstable (e.g. a mobile node with a bad wireless connection). Even if HBM includes redundancy and failure discovery mechanisms, instability must be taken into account when creating the topology. The idea is to have stable transit nodes while unstable ones are moved to the leaves of the topology. Of course the stability of a node is unknown when he first joins a session. A default (conservative) value is first assigned to the *node_stability* variable and this latter is regularly updated as time goes by.

In order to make adaptation possible, we associate a “capability” to each node. This capability has three possible values: **disconnected**, **leaf_only** (i.e. is a nonCM), **transit_possible** (i.e. is a CM). We first calculate a normalized capability, *NCap*:

$$NCap(node) = f(user\_desires, node\_stability, RP\_param)$$

where *RP_param* is a parameter specified by the RP to influence the capability of a node (e.g. if all the users choose to be **leaf_only**, then the RP can oblige some them to be transit node). Then *NCap(node)* is compared to predefined thresholds in order to determine the exact capability of the node:

if $(NCap(node) \in [0; \alpha])$, then the node is disconnected (exceptional if α is small);
 if $(NCap(node) \in [\alpha, \beta])$, then the node has capability “leaf_only” (nonCM);
 if $(NCap(node) \in]\beta, 1])$, then the node has capability “transit_possible” (CM);

This is a lightweight mechanism as the RP already keeps per-node state information. It only adds four variables: the *user_desires*, the *node_stability* (dynamically updated), the *RP_param* and the node capability.

3.3 Possible Topologies

So far OM work essentially focussed on trees (e.g. [3]). In our work we consider several potential topologies, each of them having distinctive features (figure 1):

bus: serial connection of all the nodes.

tree: several kinds of trees are possible, like Shortest Path Trees (SPT) and Minimum Spanning Trees (MST). A SPT is per-source and in case of different sources, several SPT must be created which turns out to be costly. On the opposite a MST is source independent which is an asset with (n, m) group communications.

ring: solution of the “traveling salesman problem”. The topology is source independent.

star: all the nodes are connected around a central node.

sun: a “sun” is a “star” with a non null diameter. It is therefore composed of an internal “ring” with peripheral “solar beams”.

hybrid: hybrid topologies are possible that mix for instance the “tree” and “sun” solutions.

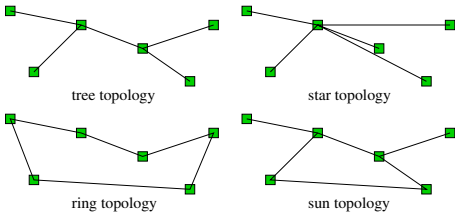


Fig. 1. Some possible overlay topologies.

Choosing one of these topologies has serious impacts on robustness and performances. To analyze them we wrote a simulator. It takes in input a randomly and homogeneously distributed³ set of nn nodes. A topology solver is run, creating MST and Ring topologies. Each topology is then analyzed, failures introduced, and statistics gathered. Experiments are repeated 30 times for each nn .

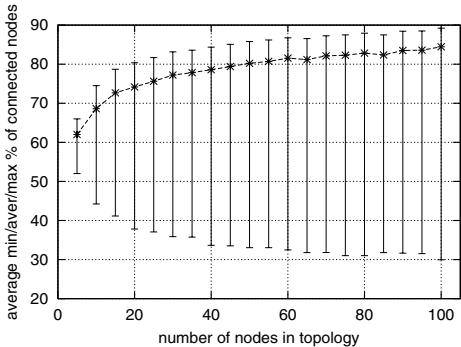


Fig. 2. Impacts of a single node failure on the connectivity of a MST.

Robustness in Front of a Node Failure. We simulated the impacts of a single node failure on the connectivity when the OM topology consists in a Minimum Spanning Tree (MST). Results are shown in Figure 2. For each value of nn , we successively turn down each node. We then measure the number of hosts still connected, cn , and plot the average $\{min/aver/max\}$ values of cn .

This experiment shows that *with a Minimum Spanning Tree, a single node failure can easily partition the whole OM topology*. If on average 62 to 84% of nodes remain connected, this value can be as low as 30%. On the contrary with a ring a single failure does not partition the topology.

³ In case of non-homogeneous node distributions, e.g. to simulate the impacts of a trans-atlantic line, the Traveling Salesman solver that creates the ring topology must be modified to take it into account. [9] page 445 gives such an algorithm.

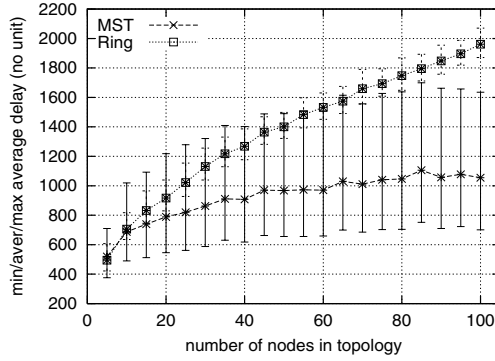


Fig. 3. Average delay for the Minimum Spanning Tree (MST) and Ring OM topologies.

Performances in Terms of Delay. [3] introduces several metrics to appreciate the quality of an overlay topology and which can be used during the topology creation process. In this section, we only focus on average group-shared delays [11].

This latter, for a given set of nn nodes and an OM topology, is given by:

$$aver\_delay(nn, topo) = mean_{all\_possible\_sources} (mean_{all\_nodes \neq source} (delay(source \rightarrow node)))$$

Not surprisingly above 15 nodes, a MST has a lower average delay than a ring. Yet the MST delay range is much higher and till 65 nodes, there are situations where the MST average delay is higher than the ring average delay.

4 Related Works

Yoid [6] is another OM scheme. If Yoid and HBM both rely on a RP, many differences exist. In particular Yoid creates a tree and uses a complex algorithm to avoid the creation of loops. Yoid also assumes that all nodes are stable.

AMRoute [2] [8], developed for Adhoc networks, establishes an overlay topology for multicast communications. AMRoute distinguishes two kinds of topology: *the mesh*, a highly interconnected topology and *the tree*, a subset of the mesh, used for an efficient data delivery. If AMRoute does not try to evaluate inter-host distances, the use of an Expanding Ring Search (ERS) algorithm takes into account locality. If convenient in an Adhoc network where wireless communications enable diffusion, ERS is not feasible in the Internet unless multicast is already available!

In NARADA [3] a mesh is first created and then a Reverse-Path Forwarding algorithm (e.g. DVMRP) is run on top of it to create a SPT per source. Many differences exist with HBM: there is no global view of the topology, it requires the use of an incremental mesh improvement technique, and per-node adaptation is not possible (a problem in case of highly heterogeneous nodes).

In [7] an application layer routing architecture (ALR) is automatically created using an active network framework (ALAN). The topology creation process follows a hierarchical approach (for improved scalability) and relies on several metrics.

5 Conclusions

This work introduces an overlay multicast solution, HBM, which offers a group communication service to all hosts, even those located in a site where inter-domain multicast is not available. It discusses the issues raised by the creation and the management of this overlay topology. We argue that a centralized solution, where the group membership is known by a Rendez-vous Point (RP), has many benefits over distributed solutions. Having a centralized topology management is *simple* (no coherency problem), *efficient* (the RP can create an optimal topology with respect to the distance database accuracy) and *takes advantage of known node features* during the topology creation process (e.g. to have stable transit nodes). We also argue that having additional redundant connections, even if it introduces additional traffic and loops, is important in front of the intrinsically fragile nature of an overlay topology.

References

1. K. Almeroth. The evolution of multicast: from the mbone to inter-domain multicast to internet2 deployment. *IEEE Network, Special Issue on Multicasting*, January 2000.
2. Bommaiah, A. McAuley, R. Talpade, and M. Liu. *AMRoute: Adhoc multicast routing protocol*, August 1998. work in progress; <draft-manet-amroute-00.txt>.
3. Y-H Chu, S. Rao, and H. Zhang. A case for end system multicast. In *ACM SIGMETRICS*, June 2000.
4. T. Cicic and H. Bryhni. *Multicast-unicast reflector*, January 2000. unpublished document, available at URL: <http://www.ifi.uio.no/~tarikc>.
5. C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the ip multicast service and architecture. *IEEE Network*, pages 78–88, January 2000.
6. P. Francis. *Yoid: extending the multicast internet architecture*, September 1999. Unrefered Report; <http://www.yallcast.com>.
7. A. Ghosh, M. Fry, and J. Crowcroft. An architecture for application layer routing. In *2nd Int. Working Conf. on Active Networks (IWAN2000)*, October 2000.
8. M. Liu, R. Talpade, and A. McAuley. Amroute: Adhoc multicast routing protocol. Technical Report TR 99-1, CSHCN, 1999.
9. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: the art of scientific computing, second edition*. Cambridge University Press, 1992.
10. V. Roca. *Host Based Multicast: an alternative group communication service*, July 2000. Tutorial presented at the “Ecole d’été des réseaux hauts débit et multimédia” (RHDM’00).
11. L. Sahasrabuddhe and B. Mukherjee. Multicast routing algorithms and protocols: a tutorial. *IEEE Network*, pages 90–102, January 2000.

A An Algorithm for the Addition of Redundant Virtual Links (RVL)

This section describes a scheme to add RVL into the logical topology to decrease the probability of partitioned topology after a single node failure. The RP defines how many redundant links to create by using the following algorithm:

```
// initialization
acceptable_threshold = 0 // or another value in [0; 1[
AddRedundantLinks(virtual_topo, all_members_in_virtual_topo);

// recursive solver
AddRedundantLinks (topology, group)
{
    if (proba(partitioned_topo, 1 failure) <= acceptable_threshold)
        exit;           // solution found, no need to go further
    if (number of members in group <= 2)
        return;
    find the set N of farthest CM nodes in logical group;
    foreach (2 nodes, N0 and N1, in N) {
        1. split the logical topology into two subgroups (subg1, subg2)
           such that each subgroup includes either N0 or N1 and all the
           nodes that are closer to it than to the other in the PHYSICAL
           topology (not necessarily on the virtual topology!)
        2. calculate the probability of partitioned topology in the case
           of a node failure before and after adding the RVL between N0
           and N1.
        if (new_proba(partitioned_topo, 1 failure) >=
            previous_proba(partitioned_topo, 1 failure))
            return;      // do not add this link
        else {
            Add this redundant link N0 <--> N1 to topology;
            AddRedundantLinks(topology, subg1); // continue with subg1
            AddRedundantLinks(topology, subg2); // continue with subg2
        }
    }
}
```

Figure 4 (a) describes physical topology and figure 4 (b) the initial OM tree that has been created. In this tree the two farthest nodes are G and D. The subgroup (sub1) for D is A,B,C,E and F. The subgroup (sub2) for G is H and I⁴ as described in figure 4 (c). The probability of partitioned topology without and with the addition of the RVL $G \rightleftharpoons D$ are 1 and $3/8$ respectively. Therefore, the redundant virtual link $G \rightleftharpoons D$ is accepted. As this probability is still greater than zero, the RP repeats this algorithm on each of the two subgroups.

For sub1, there are two pairs of farthest nodes; D with F and D with E as shown in figure 4 (d). As adding the RVL $D \rightleftharpoons F$ does not reduce this probability,

⁴ I is closer to G than to D on the physical topology!

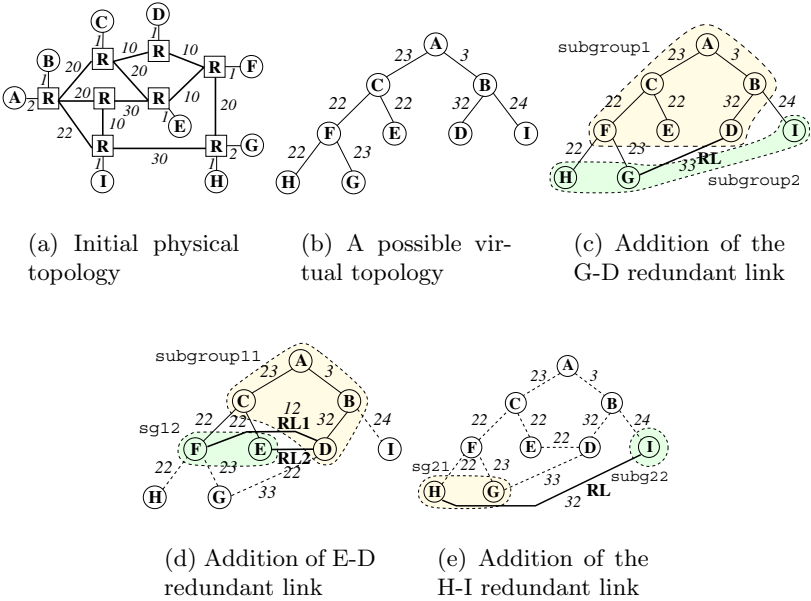


Fig. 4. Addition of Redundant Virtual Links (RVL), an example.

it is not accepted. On the opposite adding RVL $D \rightleftharpoons E$ reduce this probability from $3/8$ to $2/8$ and is accepted. The same algorithm is then executed on each subgroup of sub1, namely sub11 and sub12. As the probability cannot be further reduced, the analysis of sub1 is finished.

For sub2, the algorithm leads to the addition of link $H \rightleftharpoons I$ as shown in figure 4 (e). As the new probability reaches zero, the analysis of sub2 finishes.

At the end, with three RVL, $G \rightleftharpoons D$, $E \rightleftharpoons D$, and $H \rightleftharpoons I$, the probability of having a partitioned topology after a single node failure is null.

In the previous example, the algorithm runs until the probability of failure turns zero. In practice, we can define an “*acceptable partitioning probability*”: $acceptable\_threshold \geq 0$, and stop the solver when this value has been reached.

An Evaluation of Shared Multicast Trees with Multiple Active Cores

Daniel Zappala and Aaron Fabbri*

Department of Computer Science, University of Oregon,
Eugene, OR 97403-1202, USA [zappala|fabbri]@cs.uoregon.edu

Abstract. Core-based multicast trees use less router state, but have significant drawbacks when compared to shortest-path trees, namely higher delay and poor fault tolerance. We evaluate the feasibility of using multiple independent cores within a shared multicast tree. We consider several basic designs and discuss how using multiple cores improves fault tolerance without sacrificing router state. We examine the performance of multiple-core trees with respect to single-core trees and find that adding cores significantly lowers delay without increasing cost. Moreover, it takes only a small number of cores, placed with a k -center approximation, for a multiple-core tree to have lower delay than a single-core tree with optimal core placement. We also find that traffic concentration is avoided as long as the load is spread among a set of cores. These results indicate that shared trees with multiple active cores are a viable alternative to shortest-path trees.

1 Introduction

Multicast routing protocols are built using two basic types of trees: single-source shortest-path trees and shared core-based trees. In each case, a set of senders wants to deliver data to a set of members, known as the multicast group. With shortest-path trees, a separate tree is built for each source, using the least-cost paths between the source and the members. With a shared tree, one tree is built for the entire group and is shared among all the senders. Core-based trees are a simple way to build shared trees; a single router is chosen as the core, and a shortest-path tree is built from the core to the members. Senders transmit data toward the core until it reaches the tree.

Shared trees have a significant advantage over single-source trees in that only one routing table entry is needed for an entire group, instead of one per source. Hence, BGMP [1] uses shared trees for interdomain multicast to conserve state within the Internet backbone.

Despite this advantage, core-based trees have a number of drawbacks relative to shortest-path trees. Foremost among these is that core-based trees on average impose a higher delay between a source and the group members [2]. This is because packets often must travel first to the core and then to the group members,

* This work was supported in part by the National Science Foundation under Grant ANI-9977524 and NCR-9714680.

and the core may not be along the shortest path to each member. In addition, the core is a single point of failure; although PIM [3] uses a list of backup cores [4], members may experience significant additional delay when a core fails. Finally, using core-based trees may cause *traffic concentration*, in which some links in the network are much more heavily utilized than others [2,5].

Surprisingly, very little research has been conducted to study the possibility of using multiple cores to ameliorate these problems. The designers of both PIM and CBT [6] considered using multiple cores, but chose to use a single core early in the design stage. OCBT [7] uses a hierarchy of cores, in which cores at lower levels join to their parent in the higher level, forming a tree. OCBT's use of multiple cores helps to avoid looping problems that were present in initial designs of CBT. However, because each of the cores cooperate to form a single tree, this structure does not behave any differently than a single-core tree with respect to delay, fault tolerance, or traffic concentration.

In this paper we demonstrate the promise of building shared multicast trees with multiple, *independent* cores. Each core is the center of a separate multicast tree, and there is no coordination or dependencies among cores. This design improves the fault tolerance of the shared trees and can significantly improve performance.

Our results show that using multiple cores decreases the delay experienced by group members, since there is a greater possibility that each member will have a core near its shortest path. Furthermore, we achieve the surprising result that a small set of cores placed with a k -center approximation can produce a tree with lower delay than a single-core tree with optimal core placement. Finally, we show that multiple cores can spread the load of multicast traffic, eliminating the problem of traffic concentration. Based on these results, we conclude that shared trees using multiple cores are a significant improvement over single-core trees and thus a viable alternative to shortest-path trees.

We begin by examining two alternatives for multiple-core trees and describe how these protocols can be implemented. Then we present the results of an extensive simulation study examining the performance of these designs with respect to delay, cost, and traffic concentration.

2 Multiple-Core Designs

We consider two possible designs for multiple core trees and their implications for designing a multicast routing protocol. Both of these basic designs result in at most one copy of each packet being delivered to the group members.

2.1 Alternative Designs

Our multiple-core designs share some basic functionality, namely each core is the root of its own bidirectional, shared multicast tree, spanning some or all of the group members. Senders that are not group members transmit packets toward the core until they reach a router that is part of the bidirectional tree. Using this as a foundation, we explore these two designs:

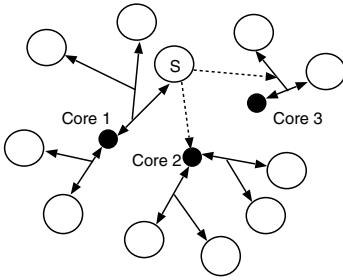


Fig. 1. Senders-To-All

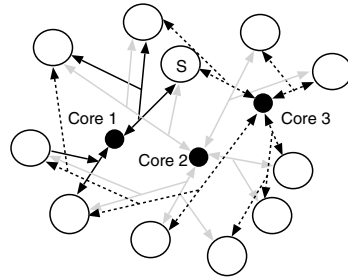


Fig. 2. Members-To-All

1. **Senders-To-All.** Each sender transmits data to all the cores; members join to only one of the cores. Fig. 1 illustrates how the Senders-To-All protocol works. In this example, there are three cores, each of which is the center of a separate, bidirectional tree connecting a subset of the group members. A sender, marked by *S*, is also a member and has joined to core 1. When it transmits a packet, it sends 3 copies, one toward each core, until they reach some router on the tree for that core. To receive packets, a member chooses a core and joins that core's shared tree.
2. **Members-To-All.** Each sender transmits data to just one of the cores; members join all of the cores. Fig. 2 illustrates how the Members-To-All protocol works. As with the previous example, there are three cores, but this time all members have joined to all the cores. In effect, this creates n redundant trees, and the sender chooses only one of them to transmit its data. In this example, sender *S* is also a member so it is joined to all of the trees. When it transmits a packet, it can send it on any one of the three trees, and can in fact choose a different tree for each packet. Likewise, different senders can utilize different cores.

We are also investigating a third design, in which the senders and members both use only one core, and the cores distribute multicast packets among them. Distribution among cores could be done using a spanning tree, a ring, or some other structure. We do not consider this design in this paper.

2.2 Senders-to-All Advantages

The Senders-To-All design has several significant advantages when compared to Members-To-All. First, Senders-To-All will on average use less router state, which we define as the number of routing entries per group at a given node. For both designs there is one tree per core, but for Members-To-All, each tree connects all of the members. Thus, for Members-To-All each router is likely to have one entry per core for each group, particularly as the group size grows. For Senders-To-All, on the other hand, each router is likely to have only one entry per group, especially if nearest attachment is used.

The Senders-To-All design also has the advantage of giving group members control over choosing a core. With Members-To-All, the source is responsible for choosing a core, then monitoring its status so that it can switch to a new core in case of failure. With Senders-To-All members have greater flexibility, since a given core may be good for some members and bad for others, depending on its location and the status of the network. With Senders-To-All, members can react quickly to failures and can even switch cores in order to improve performance characteristics such as loss and delay.

It is also worth noting that, compared to a protocol such as PIM, both of our designs reduce the delay incurred when a core fails. Since the cores are already active (either receiving data or having members joined), the time required to switch cores after a failure is reduced. Moreover, both designs localize the recovery delay to only those senders or members who are using the failed core.

2.3 Using Multiple Cores

In order to use multiple cores, group members (or their first-hop routers) need a mechanism to discover the identities of cores and select the one they will use.

We refer to the first of these issues as *core advertisement*. For single-core protocols such as PIM, the current method for advertising cores [4] is to distribute a set of candidate cores throughout the network. When a source or group member needs to send to or join a group, it selects one of these candidate cores to act as the core for the group. All of the sources and group members deterministically select the same core because PIM uses a hash function based on the group identifier.

Our multiple-core designs can use this same mechanism to distribute candidate cores and to choose a different *set* of active cores for each group. The PIM hash function produces a different ordering of cores for each group and PIM uses this ordering to select a backup core should the primary core fail. Similarly, a multiple-core protocol can use the hash function to select a set of n cores, along with backups for each of them.

Once the set of cores is known, the group members must decide which core to utilize. For the Senders-To-All design, members must decide which core to receive packets from. Likewise, for the Members-To-All design, senders must decide which core to send packets to. In most cases, choosing the nearest core should give good performance. If a core becomes congested, however, then a member or sender may want to switch to a different core.

Finally, the placement of the cores can impact the cost and delay of the multicast tree. We examine the effects of several core placement algorithms in the next section.

2.4 Implementation Details

The primary issue that must be addressed to implement our multiple core designs is packet forwarding. The current multicast routing architecture allows for only one shared tree per group. However, in our designs, there are several or many

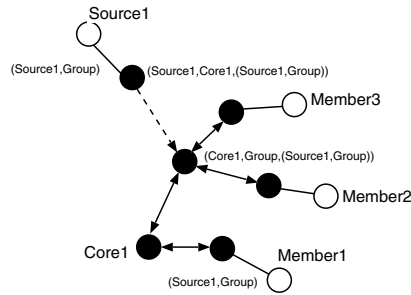


Fig. 3. Using a Bidirectional Tree at a Core

shared trees per group (one per core). These trees may overlap and thus must be identified and built individually.

To facilitate our discussion we must first explain the current types of routing entries used for multicast. Currently, a multicast routing entry may be designated using either (S, G) or $(*, G)$ where S is the source address of a multicast packet and G is the group address. If a multicast packet matches a (S, G) entry, routers assume a shortest-path tree is being used; the packet must arrive on the incoming interface listed in the entry and is sent on all outgoing interfaces. Likewise, if a multicast packet matches a $(*, G)$ entry, routers assume a shared tree is being used and the packet is forwarded accordingly (either on a unidirectional or bidirectional shared tree).

Multiple core trees, as described above, require a new type of multicast routing entry. We call this a (C, G) entry, where C is the IP address of the core. A router that receives a multicast packet matches it against (C, G) entries the same as it would for an (S, G) entry. However, if a match is found, then the packet is forwarded on a bidirectional tree; that is, it is sent on all interfaces listed except for the interface on which it arrived. In practical terms, this change may only require a few bits in the routing entry to specify how forwarding should be performed.

Fig. 3 illustrates how packets are forwarded on a multiple-core tree using this new type of routing entry. When a source sends a packet to the group, its first-hop router encapsulates it and unicasts the packet toward its nearest core. When it reaches a router on the core's bidirectional tree, this router removes the original packet and then re-encapsulates it, this time using the core's address as the source address. This packet is multicast along the bidirectional tree until it reaches the leaf routers. These routers remove the original packet and deliver it to their local members. Note that this process requires packets to be encapsulated twice. If a sender is also a member, then only one encapsulation is needed because the unicasting step is eliminated.

The steps for receiving packets depend on whether the Senders-To-All or Members-To-All design is being used. For Senders-To-All, a member's first hop router chooses one core and joins the bidirectional tree for that core. For Senders-To-All, the first-hop router joins the shared trees for all of the cores. These trees

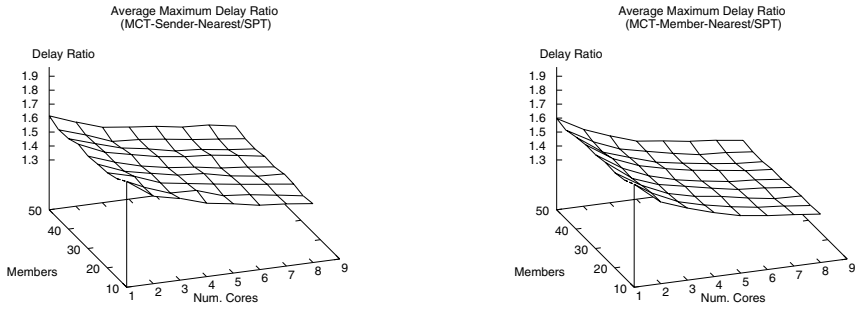


Fig. 4. Delay for Random Core Selection and Nearest Attachment: Senders-To-All (left) and Members-To-All (right)

are joined in the usual manner, that is, by sending a separate join message toward the core for each tree.

3 Simulation Study

We evaluate the feasibility of multiple-core trees by comparing their performance to single-core trees, as has been done in several previous studies comparing tree types [2] and core selection algorithms [5]. The factors in our experiment include group size (from 5 to 50), core selection (random and dominating set), and core attachment (random and nearest). The metrics we evaluate include cost, delay and traffic concentration. We report ratios to the corresponding SPT metrics, so that we can compare results across different graphs and groups.

3.1 Experiment Results: Delay

For these experiments, we use a set of 10 flat, random graphs of 50 nodes each, using the Waxman model [8] within the GT-ITM topology generator [9]. All edges have unit weights and the average node degree is near 4. For each graph, we generate 50 random groups and measure the delay experienced by group members. We define *delay* as the number of links traversed between one sender and one member in the group. We calculate the maximum delay for each sender, then average these numbers to find the average-maximum for the group.

We find that both Senders-To-All and Members-To-All can significantly reduce the delay experienced by group members when nearest attachment is used. As shown in Fig. 4, the delay decreases dramatically as the number of cores increases. This is because there is a greater chance that the member (or sender) will choose a core that is close to the shortest path.

Particularly interesting in these results is that, for nearest core attachment, most of the benefits are seen with only 5 cores, after which the graphs are mostly

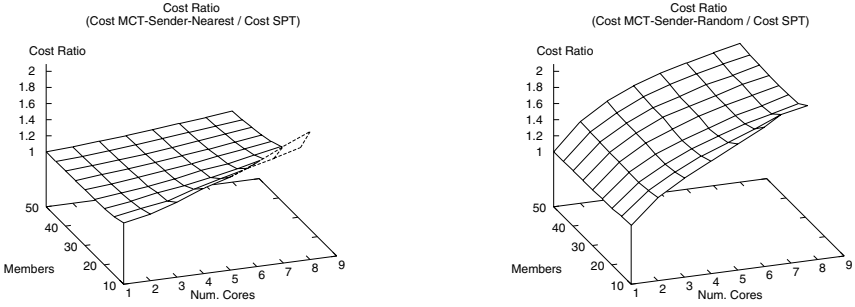


Fig. 5. Cost for Senders-To-All with Random Core Selection: Nearest Attachment (left) and Random Attachment (right)

flat. Thus only a small number of nodes – about 10% – need to be used as cores for a group.

Our results also show that the Members-To-All design is tolerant of members choosing distant cores: with random attachment delay increases only slightly as more cores are added. The Senders-To-All design, on the other hand, can suffer from large maximum delays when using random attachment. In the extreme case, with both a large group and a large number of cores, there is good chance that a member will choose to use some core that is distant from both itself and the sender. With the Senders-To-All design, the sender must transmit to all cores, so it will use the distant core and incur a large maximum delay.

3.2 Experiment Results: Cost

To evaluate cost, we use the same experiment setup as for delay. We define *cost* as the number of links in a tree, representing the bandwidth consumed by one packet transmission. We calculate cost separately for each sender, then average these costs together to find the average cost for the whole group. Note that for Senders-To-All, we count each link for each packet sent; thus if three packets are sent to three different cores, it may be possible for one link to be counted three times.

We find that the Members-To-All design performs better than Senders-To-All with respect to cost. Sending packets using Members-To-All does not consume much more bandwidth than a shortest-path tree. The cost ratio is nearly flat and close to 1 for both random and nearest attachment. In fact, the Members-To-All design actually performs slightly better than a single-core tree in this regard. This is because a sender is able to choose a core whose performance is very close to that of a shortest-path tree.

Senders-To-All also performs well with regard to cost as long as nearest attachment is used (see Fig. 5). The only exception is that the cost increases for

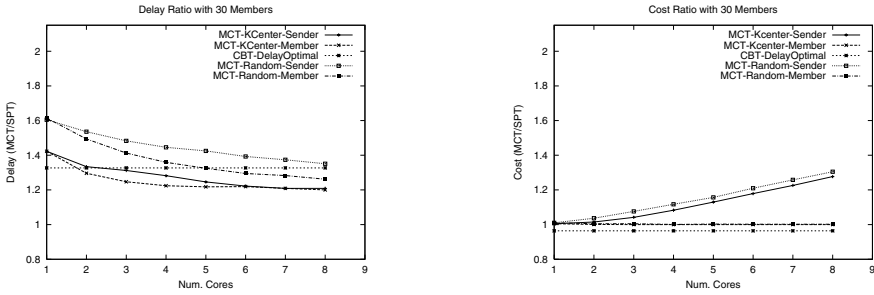


Fig. 6. K -Center Placement and Nearest Attachment: Delay Ratio (left) and Cost Ratio (right)

small groups with many cores, due to the number of copies generated on links close to the sender.

Senders-To-All does not perform well with random attachment; as shown in Fig. 5 the cost ratio doubles as the number of cores increases. This happens because the sender generates a separate packet for each core and transmits each copy on a separate, but possibly overlapping tree. The copies will often travel some of the same links, multiplying the amount of bandwidth consumed by the sender.

3.3 Experiment Results: Core Selection

To examine the effects of core selection, we used both a *dominating set* algorithm and a k -center algorithm. A dominating set is a subset of the nodes in the graph such that all nodes are within n hops of this set. A k -center algorithm fixes the number of cores to be equal to k , then tries to place them so as to minimize the distance from all nodes to the cores. Both finding a minimal dominating set and choosing an optimal placement of cores is NP-hard. We use approximation algorithms based on node degree, and our experiments indicate these are good approximations in random graphs.

We find that both dominating set placement and k -center placement improve the delay and cost ratios for multiple-core trees. Delay is reduced by 10 to 20% for the different designs; the cost ratio is close to 1 already so the improvement is slight. Moreover, as the number of cores increases, multiple core trees using both random core selection and k -center placement can have lower delay than a single-core tree with optimal core placement! Fig. 6 shows these results. In this same figure we also show the cost ratio for the corresponding experiment. Again, K -center placement is an improvement over random core placement. The single-core tree with optimal core placement retains the lone advantage that it can have lower cost than a shortest-path tree. This is in keeping with the result from Calvert et al. [5] in which optimal core placement is the only placement mechanism they study where single-core trees have lower cost than shortest-path trees.

3.4 Experiment Results: Traffic Concentration

For traffic concentration, we followed the methodology used by Wei and Estrin.[2] For a given graph, we generate 300 groups and construct their multicast trees. Then, for each group, we transmit a single packet from each source. As each packet is sent, we count the number of times each link in the network is traversed. With multiple-core trees, it is possible that a given link is traversed more than once by a single packet, due to encapsulation.

Our results show that as long as random core selection is used, both Members-To-All and Senders-To-All do not suffer from traffic concentration, regardless of the number of cores. This confirms the results of Calvert et al. [5] indicating that traffic concentration is only observed when a small number of cores is used across all groups. With random core selection, each group uses a different set of cores, so the traffic is spread throughout the network.

We do observe traffic concentration when using k -center placement, because in this case all groups use the same set of cores. In this situation, the effects of traffic concentration appear when there are fewer than 4 cores. As the number of cores increases beyond this amount, traffic concentration disappears.

4 Conclusions and Future Work

Our results indicate that multiple-core trees are a feasible alternative to shortest-path trees. They can have lower delay than trees using a single core and cost comparable to shortest-path trees. In addition, multiple-core trees do not suffer from traffic concentration, as long as a reasonably large set of candidate cores is used. An ISP may also use a k -center algorithm to choose a static set of cores; this can reduce delay and will avoid traffic concentration as long as a large enough set is used.

We are particularly interested in the Senders-To-All variant since it uses less router state and provides group members with more flexible control when reacting to failed cores and congestion. In most situations, the delay and cost of the Senders-To-All design are close to that experienced with shortest-path trees, as long as nearest attachment is used. In this case, its primary drawback is its higher cost with small groups and large numbers of cores. We are exploring ways to reduce cost in this situation. Costs can increase when members choose distant cores, but a natural disincentive (higher delay) or policy rules will likely prevent this from happening.

The Members-To-All variant has better performance in terms of cost and delay, and is also tolerant of group members choosing distant cores. It is less flexible with regard to fault tolerance and uses more router state, but we still consider it a viable option for multicast.

We are continuing to explore the design of multiple-core multicast protocols, particularly with respect to single-source multicast. We are also intrigued by the possibility of multiple core trees that use core distribution and are actively investigating the cost and delay attributes of these structures as well.

Acknowledgments

The authors would like to thank Virginia Lo for substantial contributions during the development of this work, as well as Krishnan Sivaramakrishna Iyer for help developing the dominating set approximation algorithm.

References

1. S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, and M. Handley, "The MASC/BGMP Architecture for Inter-domain Multicast Routing," in *ACM SIGCOMM*, August 1998.
2. Liming Wei and Deborah Estrin, "The Trade-offs of Multicast Trees and Algorithms," in *1994 International Conference on Computer Communications Networks*, September 1994.
3. Stephen Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei, "An Architecture for Wide-Area Multicast Routing," in *ACM SIGCOMM*, August 1994.
4. Deborah Estrin, Mark Handley, Ahmed Helmy, Polly Huang, and David Thaler, "A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing," submitted to *ACM/IEEE Transactions on Networking*.
5. K.L. Calvert, E.W. Zegura, and M.J. Donahoo, "Core Selection Methods for Multicast Routing," in *International Conference on Computer Communications Networks*, September 1995.
6. A. J. Ballardie, P.F. Francis, and J. Crowcroft, "Core Based Trees," in *ACM SIGCOMM*, August 1993.
7. C. Shields and J.J. Garcia-Luna-Aceves, "The Ordered Core Based Tree Protocol," in *IEEE INFOCOM*, 1997.
8. Bernard M. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, December 1988.
9. Ken Calvert and Ellen Zegura, "Georgia Tech Internetwork Topology Models," <http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html>.

QoS Routing Protocol for the Generalized Multicast Routing Problem (GMRP)

Hatem Bettahar and Abdelmadjid Bouabdallah

Heudiasyc – UMR CNRS 6599
Université de Technologie de Compiègne
BP 20529 – 60205 COMPIEGNE cedex FRANCE
hatem.bettahar@utc.fr
<http://www.hds.utc.fr/~hbettaha/>

Abstract. In this paper we study the importance of group characteristics in multicast communications, and we present a generalized multicast routing schema called GMRP (Generalized Multicast Routing Problem). The goal of GMRP is to provide an efficient multicast routing approach based on the group characteristics. As a case study we considered the group dynamism where we distinguish between fixed and dynamic receivers, we present an efficient routing algorithm called Maximum Degree Minimum Delay Algorithm that takes advantage of this group characteristic.

Key words: Multicast routing ,group characteristics, QoS, minimum delay protocol.

1 Introduction

Multicast routing has widely contributed in the deployment of multimedia group applications on the Internet such as teleconferencing, tele-education and computer supported collaborative work [1] [2] [6]. Such applications often have stringent quality of service (QoS) requirements. Unfortunately, most current deployed multicast protocols are based on the shortest-path algorithms and hence they have no QoS capabilities [3]. In addition these protocols construct multicast trees without caring about the type of the group itself. We believe that efficient trees may be obtained only if we consider the different characteristics of the group. In current multicast protocols, the only group characteristic that is considered as a major factor is the dispersion of the group members, as a result, two modes are defined: the Sparse mode for groups with few member separated by large WANs and the dense mode for groups with lots of receivers concentrated in the same area. Different protocols are used in the two modes to allow an efficient multicast routing [5]. In the same way, many other group characteristics should be considered in the multicast routing approach such as the group dynamism (fixed/dynamic members), the way members join the group (join-only, join-leave) the size of the group and its composition (source-only,source-receiver, receiver), the periodicity of the group (permanent, periodic, temporary) and

QoS requirements of the group (delay constraint, bandwidth constraint, homogeneous/heterogeneous QoS requirements for different members).

In this paper we propose to generalize the multicast routing problem to include different multicast group characteristics in the routing scheme. We call this problem the Generalized Multicast Routing Problem.

To prove the importance of group characteristics in multicast routing, we present a case study of the group dynamism where we distinguish between fixed and dynamic receivers, we present an appropriate algorithm called Maximum Degree Minimum Delay Algorithm which is a greedy algorithm with a cost function that combines link costs and nodes degrees. By simulating this algorithm we show that we can take advantage of this group characteristic to provide more efficient routing algorithms. We believe that this should be true for any other group characteristic.

The rest of this paper is organized as follows: in section 2 we define the Generalized Multicast Routing Problem by presenting the network model, multicast group characteristics and a formulation of the GMRP. In section 3 we discuss a case study of the GMRP by taking the group dynamism as an example of important group characteristic, we present an appropriate routing algorithm, called Maximum Degree Minimum Delay Algorithm (MDMDA), for building efficient multicast source trees in this case. In section 4 we describe the simulation model that we use to compare multicast routing algorithms and we discuss the results we obtained for our routing algorithm. This paper is concluded in section 5.

2 GMRP: The Generalized Multicast Routing Problem

In this section we present the different component of the Generalized Multicast Routing Problem. We first describe the network model and different link and node parameters. We then enumerate different multicast group characteristics and we describe how they influence the efficiency of the multicast routing schema. Finally we give a general definition of the Generalized Multicast Routing Problem.

2.1 Network Model

A network is modeled by a connected graph $N = (V, E)$, where V is a set of nodes and E is a set of directed links. To each link $e = (u, v) \in E$ we associate three metrics: a cost c_e a Delay d_e and an available bandwidth b_e . The link cost may be hop count, a monetary cost or any other cost function. The link delay includes queuing, transmission and propagation delays of a packet. For a path $P(u, v)$ the cost $c_{P(u, v)}$ is the sum of costs of the path links, the delay $d_{P(u, v)}$ is the sum of delays of the path links and the available bandwidth $b_{P(u, v)}$ is the minimum available bandwidth on the path links.

$$c_{P(u, v)} = \sum_{e \in P(u, v)} c_e$$

$$d_{P(u,v)} = \sum_{e \in P(u,v)} d_e$$

$$b_{P(u,v)} = \min_{e \in P(u,v)} b_e$$

A Multicast Tree $T(G)$ is a tree spanning all members of the multicast group G . The set of sources S of the group G is not necessarily included in the tree, a source may not be a member of the group. During the joining phase a receiver r may specify a subset S_1 of S from which he will receive multicast traffic. The total cost of the tree $T(G)$ is the sum of costs of links in the tree. The delay for the receiver r is the maximum delay on paths $P(s_i, r)$ with $s_i \in S_1$. The available bandwidth used by the receiver r is the minimum available bandwidth on paths $P(s_i, r)$ with $s_i \in S_1$. The total available bandwidth on the Shared multicast Tree $T(G)$ is the minimum available bandwidth between a source and a receiver.

$$b_{T(G)} = \min_{u \in G, v \in S} b_{P(u,v)}$$

The delay diameter of the tree $T(G)$ is the maximum delay between a source and a member of the group.

$$Diam(T(G)) = \max_{u \in G, v \in S} d_{P(u,v)}$$

In section 3 and 4 we consider only source based trees to simplify our case study and simulation results but all what we present can be extended to shared trees.

2.2 Group Characteristics

A multicast group G is a set of nodes participating in the same multicast session, we note $G = g_1, g_2, \dots, g_n \subseteq V$ with $n = |G| \leq |V|$. G is identified by a unique class D address. In current deployed multicast protocols we can distinguish two types of routing depending on group modes: *Dense* and *Sparse*, these two modes are based on some group characteristics such as dispersion of members and the size of the group. In the following we discuss more characteristics that influence the multicast routing and that should be introduced in the multicast routing problem.

Size of the group : This is an important characteristic of a multicast group, the number of sources number receivers, number of source/receiver members are important factors in the routing schema.

Group Dynamism : In many current multicast groups, receivers may be *Fixed* (known, subscribed,) or *Dynamic* (visitors). This is the case for example in a tele-education session, where the audience may consist of subscribed students of the class (Fixed receivers) and other visitors (Dynamic receivers) who can occasionally join the group. We will discuss in details the influence of such distinction between fixed and dynamic receivers in section 3.

The Type of Session : A multicast session may be a *Join only* or *Join/Leave* session, in [4] the author studied the greedy and the Naive algorithms in different session types.

Duration of the group : A multicast group may be permanent, periodic or temporary group. In each case a different kind of routing should be applied to take advantage of this group characteristic and so to offer an efficient routing algorithm, for example in the case of periodic groups a prior calculation of routes as well as prior reservation may be a good routing choice. In the case of permanent groups with a majority of fixed members we can apply static routing algorithms with periodic tree update to keep the efficiency of the tree within a certain level.

QoS requirements of the group : Depending on the multicast application, the group may have stringent QoS requirement for different parameters such as end-to-end delay, bandwidth, loss rate and jitter. Members of the same group may have homogeneous or heterogeneous QoS requirement. These group specificities should be considered during path calculation.

In table 1 we summarize the above characteristics of a multicast group, we note that this list may be completed by many other group characteristics that may also influence the routing mechanism, We cite here only important ones.

Table 1. Group characteristics

Group characteristics	Definitions
Group size	Number of sources, receivers, source-receivers.
Group dynamism	% of fixed members, % of dynamic members
Session type	join only session, join/leave session
Group duration	permanent, periodic, temporary.
QoS requirement	no requirement, delay-bandwidth constraints, ...

2.3 GMRP Definition

The Generalized Multicast Routing Problem, GMRP, should be seen as a generalization of any multicast routing schema. Given a network within the description given in paragraph 2.1, and a multicast group with specific characteristics as described in paragraph 2.2, the GMRP consists on building an *efficient* tree for such a group. The *efficiency* of the tree depends on the group characteristics and on which routing parameters we want to optimize. An efficient tree may be a tree that accepts a maximum number of receivers within a certain end-to-end delay constraint, it may also be a tree with minimum total cost. It would be rather impossible to have one generic routing algorithm that produces efficient trees for all types of groups, but our goal is to build a set of generic routing mechanisms that may cover most current group models. Figure 1 presents a general view of the GMRP.

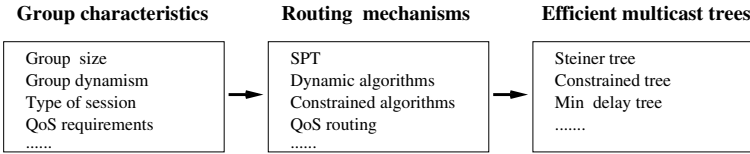


Fig. 1. the Generalized Multicast Routing Problem

3 Case Study: GMRP and Group Dynamism

In this section we discuss the influence of group dynamism in multicast routing. We first formulate group dynamism and then we present an efficient algorithm that takes advantage of this group characteristic to build the multicast tree. Our algorithm is based on the known greedy algorithm [9] .

3.1 Group Dynamism

In many multicast groups we have two types of members: *known* members (in most cases all sources and a set of receivers) and *unknown* (dynamic) members who can freely join the group at any time. To formulate this important group characteristic we propose the following notation for a multicast group G :
 $G = S \cup SD \cup DD$ with:

- G** : The multicast *Group*.
- S** : The set of *Sources* of G .
- SD**: The set of *Static Destinations* (known, fixed receivers) of G .
- DD**: The set of *Dynamic Destinations* (unknown receivers) of G

3.2 GMRP and Group Dynamism

Given a network $N = (V, E)$ which have the proprieties given in paragraph 2.1, and a multicast group G with a group dynamism characteristics as described in paragraph 3.1, the GMRP consists on constructing an *efficient* tree for such a group. In this case the efficiency of the tree may be measured by the number of dynamic destinations that the tree may accept under a certain delay or cost constraint. The general goal will be to construct a tree that connects the maximum number of destinations when minimizing the total cost and the average end-to-end delay of the tree.

We propose to split the GMRP into two sub-problems: The first sub-problem consists on constructing a *Partial Static Tree (PST)* that connects only the *Sources* and the *Static Destinations* of the group. The second sub-problem consists on connecting dynamically the set of *Dynamic Destinations* one by one in their order of arrival.

Constructing the Partial Static Tree : The *Partial Static Tree* should be constructed in a way that a maximum set of *Dynamic receivers* could join the tree later on with a minimum cost and a minimum end-to-end delay. The *Partial Static Tree* should not be optimal itself but should lead to an optimal final tree after all group members (static and dynamic) have joined the group.

Dynamic join of Dynamic Destinations : After constructing the *Partial Static Tree*, members of the *Dynamic Destinations* set should be added one by one in the order of their arrival, the join mechanisms should lead to an efficient final tree.

3.3 The Maximum Degree Minimum Delay Algorithm: MDMDA

To construct the *Partial Static Tree*, we propose to use the *Static Greedy Algorithm* [9] together with the following specific cost function: For each link $e = (u, v) \in E$ we associate the cost $COST(e)$ with:

$$COST(e) = \frac{d_e}{deg_v}$$

Where d_e is the delay of the link e and deg_v is the degree of node v , the incidence node of the link e . This cost function leads to the construction of a *Partial Static Tree* with high degree nodes while maintaining a reasonable end-to-end delay and a reasonable total cost. Such tree is efficient to add other dynamic receivers with minimum cost.

To add the dynamic destinations to the *Partial Static Tree* we propose to use the *Dynamic Greedy Algorithm* [9] together with the delay d_e as the cost function associated with each link $e = (u, v) \in E$.

Figure 2 shows the importance of having a *Partial Static Tree* with high degree nodes. Here the PST1, which has higher average degree nodes, can accept dynamic nodes 2 and 5 with less cost than that APS2 does. Even though the cost of APS1 is higher than the cost of APS2, the final tree produced by APS1 after adding dynamic destinations 2 and 5 will have a better cost.

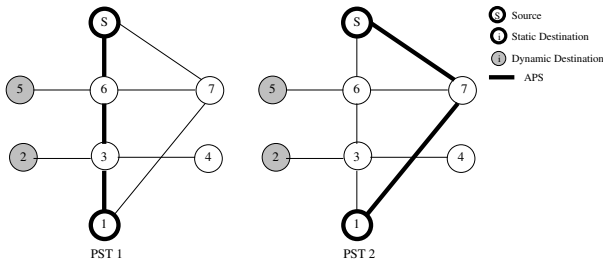


Fig. 2. The Maximum Degree Minimum Delay Algorithm: example

To simplify the algorithm presentation we will take the case of source-base trees. Table 2 presents the *Maximum Degree Minimum Delay Algorithm (MDMDA)* in this case.

Table 2. Maximum Degree Minimum Delay Algorithm (MDMDA)

Maximum Degree Minimum Delay Algorithm (MDMDA)
1. START from the source. 2. REPEAT i. SELECT among non-connected <i>Static Receivers</i> in <i>SD</i> the <i>closest</i> to the current tree, using the new cost function <i>COST()</i>. ii. JOIN the selected <i>Static Receiver</i> to the current tree. UNTIL All <i>Static Receivers</i> in <i>SD</i> are connected to the <i>Partial Static Tree</i>. 3. REPEAT for each <i>Dynamic Receiver</i> in <i>DD</i> i. JOIN this <i>Dynamic Receiver</i> to the current tree using the delay as cost function.

4 Simulations

In this section, we provide an overview of our simulation model and some of the results we obtained by comparing the Maximum Degree Minimum Delay Algorithm (MDMDA) with the classical greedy algorithm [9] (TM) and the Reverse Shortest Path algorithm (RSP) used in DVMRP.

4.1 Simulation Model

In these simulations we look for the effect of the network size and the node average degree on the end-to-end delay and the total cost for a given multicast group. We Compare Maximum Degree Minimum Delay Algorithm (MDMDA) with the classical greedy algorithm (TM) where the static version is applied for fixed members and the dynamic version is applied for dynamic members. We also compare MDMDA algorithm with the Reverse Shortest Path algorithm (RSP).

Simulations are carried over a set of Random Euclidean graphs generated using a modified version of Waxman algorithm [10] proposed by SALAMA in [8]. A Random Euclidean graph is generated by distributing nodes on an Euclidean plane uniformly and adding edges between nodes on a probalistic basis. We used graphs with 50 to 300 nodes and with average degree varying from 3 to 6, The default average degree is 4.

We used the Bi-model distribution to assign delays to edges. We assign high delay values uniformly distributed within $[90ms, 100ms]$ to 20% of links and to all the rest we assign delay values uniformly distributed between $[1ms, 10ms]$. In [7] the author suggested that Internet traffic load is skewed, with most links underutilized and a few links heavily congested, it seems logic that this can be expressed in terms of link delay, since in a highly loaded link we possibly experience a high delay.

In our simulations the multicast group represents 80% of the total size of the network and it is formed by 20% of static receivers and 80% of dynamic receivers.

Each value in our simulation is an average over 200 iterations on different generated graphs, this produces a satisfactory confidence level for our experiments.

4.2 Results

In the first series (figure 3) we compare the average end-to-end delay and the total tree cost while varying the network size from 50 to 300 nodes. Our algorithm MDMDA provide trees with total cost lightly smaller than both TM and RSP, but the more interesting gain is the average end-to-end delay. For example, with a 200 nodes network, MDMDA performs an end-to-end delay which is less by 10% than that performed by TM and by 60% than that performed by RSP.

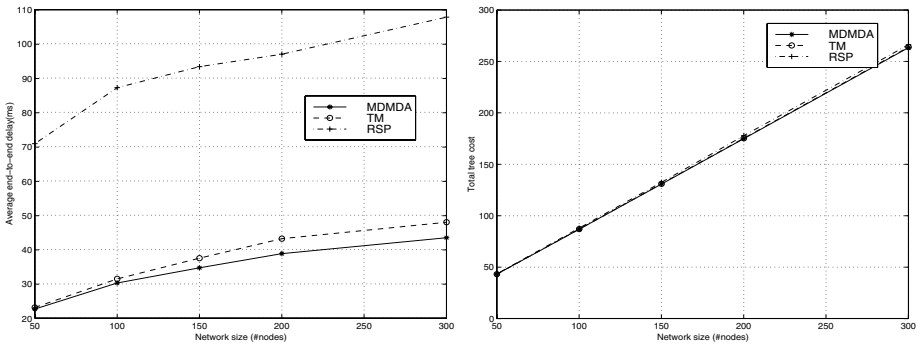


Fig. 3. Average end-to-end delay and Total tree cost as a function of the network size

In the second series (figure 4) we compare the average end-to-end delay and the total tree cost while varying the average node degree for networks of 100 nodes. For average node degrees varying between 3 and 6, The Maximum Degree Minimum Delay Algorithm (MDMDA) constructs more efficient trees compared to the greedy algorithm (TM) and the Reverse Shortest Path algorithm (RSP).

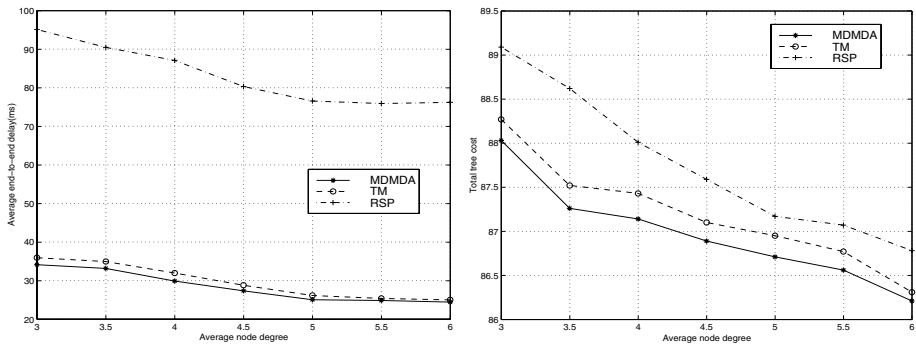


Fig. 4. Average end-to-end delay and Total tree cost as a function of the average node degree

5 Conclusion and Future Work

In this paper we formulated a generalized multicast routing schema called GMRP (Generalized Multicast Routing Problem). GMRP aims to provide an efficient routing mechanisms based on the multicast group characteristics. To prove the importance of group characteristics in multicast routing, we presented a case study of the group dynamism where we distinguish between fixed and dynamic receivers, we presented an appropriate algorithm called Maximum Degree Minimum Delay Algorithm which is a greedy algorithm with a cost function that combines link costs and nodes degrees. Simulation shows that MDMDA overperforms classical Greedy algorithm and Reverse Shortest Path Algorithm. More work should be carried to study the influence of other group characteristics on multicast routing, and to regroup all these characteristics and routing mechanisms to have a complete image of the *Generalized Multicast Routing Problem*.

References

1. J. Crowcroft, M. Handley, and I. Wakeman. *Internetworking Multimedia*. UCL Press, June 1998.
2. C. Diot, W. Dabbous, and J. Crowcroft. *Multipoint communications: A survey of protocols, functions, and mechanisms*. *IEEE Journal of Selected Areas in Communications*, pages 277–290, Avril 1997.
3. M. Doar and I. Leslie. *How bad is naive multicast routing?* *Proc. IEEE INFOCOM*, pages 82–89, 1993.
4. M. Faloutsos. *The Greedy, the Naive, and the Optimal Multicast Routing: from Theory to Internet Protocols*. PhD thesis, University of Toronto, 1998.
5. D. Kosiur. *IP Multicasting: The complete guide to interactive corporate networks*. Wiley Computer Publishing, 1998.
6. L. Wei and D. Estrin. *The Trade-offs of Multicast Trees and Algorithms*. *Proceedings of the Third International Conference on Computer Communications and Networking (IC3N'94)*, pages 17–24, 1994.

7. A. Odlyzko. *The internet and other networks: Utilization rates and their implications*. Available at <http://www.research.att.com/~amo>, 1998.
8. H. F. Salama. *Multicast Routing for Real-Time Communication on High-Speed Networks*. PhD thesis, North Carolina State University, 1996.
9. H. Takahashi and A. Matsuyama. *An approximate solution for the steiner problem in graphs*. *Math. Japonica* 24(6), pages 573–577, 1980.
10. B. Waxman. *Routing of Multipoint connections*. *IEEE Journal on Selected Areas in Communications*, vol.6, no.9, pages 1617–1622, Decembre 1988.

Feedback Scalability for Multicast Videoconferencing

Hugh Smith¹, Matt Mutka², and Lijian Yang³

¹ Department of Computer Science, California Polytechnic State University,
San Luis Obispo CA 93407, USA
husmith@calpoly.edu

² Department of Computer Science and Engineering, Michigan State University,
East Lansing MI 48824, USA
Mutka@cse.msu.edu

³ Department of Statistics and Probability, Michigan State University,
East Lansing MI 48824, USA
yang@stt.msu.edu

Abstract. Some type of transmission rate control is required in order to support multicast video applications over the Internet. Previously, we proposed a new protocol, the Layered Multicast Control Protocol (LMCP), which utilizes both the video sender and the receivers to control the rate of the video transmission. One weakness of this approach is that feedback from all receivers is required in order for the video source to determine an optimal transmission rate.

In this paper we introduce an algorithm that allows us to determine the video source's transmission rates based on feedback from a subset or sample of the receiver's feedback. As our analysis shows this algorithm allows us to support hundreds of receivers and allows the sender to determine transmission rates, which are nearly optimal. Based on the distributions analyzed, we were able to calculate transmission rates that achieved a displayable video rate within 5% of the optimal setting for 500 receivers with a worse case feedback rate at the source of 2kbps.

1 Introduction

As the Internet's networking capabilities have increased, there has been a push to utilize these capabilities for non-traditional applications. One of these non-traditional applications is multicast videoconferencing. A multicast videoconferencing application consists of a sender, transmitting video to a heterogeneous group of receivers. In a distributed environment such as the Internet, it is very difficult for this type of application to determine the availability of network resources. To perform this function some type of monitoring and control process is necessary. This process is responsible for dynamically picking transmission rates that best meets the changing needs of the set of receivers while taking into account competing applications. To simplify our discussion, we are assuming that the multicast videoconferencing traffic is separated from normal (TCP/UDP) traffic. This may be done by utilizing a separate queue (DIFFSERV) [1] in the routers for video traffic.

One of the difficulties in multicasting a video stream over the Internet is determining the rate of transmission that best meets the receivers' requirements. Previous rate-

control can be divided into two types. The first is a sender based rate-adaptation approach [2, 3] where the sender multicasts a single video signal and adjusts its transmission rate based on feedback from the receivers. This approach works well if all receivers have similar network resources, but it performs poorly if the receivers have large differences in their available network resources.

The second approach is a receiver based rate-control protocol [4, 5, 6, 7, 8]. In this solution, the single video stream is split into multiple segments in order to transmit the stream across multiple multicast groups. The receivers are then responsible for adding and dropping the multicast channels to best meet their available resources. The sender is not dynamic and transmits the video stream at pre-determined (fixed) rates. While this approach may better meet the diverse requirements of the receivers, due to its lack of sender rate-adaptation, it will not maximize the utilization of the receivers' bandwidth. An example of this approach is the Receiver-Driven Layered Multicast protocol (RLM) [7].

1.1 Layered Multicast Control Protocol (LMCP)

In [9, 10], we presented a new protocol call the Layered Multicast Control Protocol (LMCP) that combines the strengths of these previous two approaches. More specifically, this protocol utilizes both the multiple channel transmission concept from the receiver-based approach (RLM) and the transmission rate adaptation concept from the sender-based approach. The receivers not only perform the basic RLM protocol, they also determine their available bandwidth and provide this as feedback to the sender. The sender processes the receivers' feedback to determine the optimal rate of transmission for each of the multicast layers. These transmission rates are optimal in the sense that they maximize the displayable video at the receivers. In [10], we analyzed four different maximization algorithms for setting the video source's multicast transmission rates and show that our protocol significantly improves the amount of video displayed. In [9], we extend the LCMP by providing a router-based technique that allows each receiver to determine its feedback rate based on the available bandwidth on the path between the video source and the receiver. We call this feedback rate the receiver's bottleneck rate since it is the minimum (or bottleneck) rate on the source-receiver path.

The LMCP video source's control algorithm picks transmission rates, which maximizes the displayable video for the receivers as an aggregate and not any individual receiver. In order to show the effectiveness of this approach in [10], we introduced the metric *percentage-used*. This metric is calculated as the sum of the bandwidth-received for all N receivers divided by the sum of the total possible received. The higher the *percentage-used* the better, with a *percentage-used* of 1 meaning all receivers are receiving at their maximum rate. The calculation of this metric is:

$$\text{Percentage-used} = \frac{\text{Total\_received}}{\text{Total\_possible}} = \frac{\sum_{i=1}^N (r_i - (r_i - t_j))}{\sum_{i=1}^N r_i} = \frac{\sum_{i=1}^N k_i}{\sum_{i=1}^N r_i} \quad (1)$$

Where N is the total number of receivers, $r_i \in R$ is receiver i 's bottleneck rate and $t_j \in T$ where T is the video sources set of transmission rates and t_j is the maximum transmission rate such that $t_j \leq r_i$ and $k_i = r_i - (r_i - t_j)$.

One weakness with the LMCP is that as the number of receivers grows into the hundreds, the amount of feedback arriving at the sender will increase beyond the sender's ability to receive and process the information. There are two approaches for dealing with large amounts of feedback at the video source. First, the control period, which is the amount of time between successive runs of the sender's control algorithm, may be lengthened. This will allow the receivers to increase the time they wait before transmitting their feedback packet and will decrease the overhead of the feedback on the sender. The downside to this approach is that it reduces the responsiveness of the control protocol to changes in available bandwidth. The second approach is to reduce the number of receivers providing feedback in a control period. In this approach, a random subset of the receivers generate and transmit their feedback during a control period. The sender uses this subset of feedback in order to calculate its transmission rates. In this paper we develop an algorithm for implementing this type of approach through statistical sampling and discuss the accuracy of this approach.

The remainder of the paper is organized as follows. In the next section we introduce a statistical sampling algorithm for determining the video source's transmission rate. We then provide analytical results of this new approach and our conclusions.

2 Statistical Sampling Algorithm

As we mentioned earlier, one of the weakness of the LMCP is that as the number of receivers grows into the hundreds, the amount of feedback arriving at the sender will increase beyond the sender's ability to receive and process the information. In this section we are interested in determining the number of receivers which must be sampled in order to allow the video source to determine transmission rates which achieve a *percentage-used* to within an error of ϵ and a confidence of $1-\alpha$ of the true *percentage-used*. By true *percentage-used* we mean the *percentage-used* achieved when the video source runs its maximization function using feedback from all receivers. Stating this formally we are looking for the smallest sample size, n , such that $P(|P - \hat{P}| > \epsilon) \leq \alpha$, where P is the true *percentage-used* and $\hat{P}$ is the *percentage-used* achieved using only a subset of the receivers' feedback.

One difficulty in calculating this n is the independence of the two variables, k_i and r_i , in the calculation of *percentage-used*. In order to analyze the independence of these two variables we ran multiple simulations using receiver bandwidth distributions ranging between highly clustered to randomly distributed. We then calculated the correlation coefficient, ρ , between the two variables, k_i (the amount of video displayed at receiver i) and r_i (the bottleneck rate on the path between the video source and receiver i), used in equation 1. The calculation of ρ was based on Pearson's product moment correlation coefficient as given in [16]. Our results showed that is a very

high correlation between these two variables regardless of the receiver's bandwidth distribution.

As an alternative to maximizing *percentage-used*, the sender's control algorithm may minimize the *percentage-wasted*; where *percentage-wasted* = 1 - *percentage-used* and is calculated as:

$$\text{Percentage-wasted} = \frac{\text{Total\_Wasted}}{\text{Total\_possible}} = \frac{\sum_{i=1}^N (r_i - t_j)}{\sum_{i=1}^N r_i} = \frac{\sum_{i=1}^N w_i}{\sum_{i=1}^N r_i} \quad (2)$$

Where $r_i \in R$ is receiver i 's bottle neck rate and $t_j \in T$ is the maximum transmission rate such that $t_j \leq r_i$ and $w_i = r_i - t_j$.

We may then restate the problem as looking for the smallest n such that $P(|W - \hat{W}| > \epsilon) \leq \alpha$, where W is the true *percentage-wasted* and $\hat{W}$ is achieved using only a subset of the receivers' bottleneck rates. Based on our analysis we found that the correlation coefficient, ρ , for the two variables w_i and r_i is significantly less. Therefore, we may assume that the distribution of *percentage-wasted* is asymptotically normal for sufficient size n .

2.1 Derivation of Our Sample Size n

In order to save space we have omitted the complete derivation of n , our sample size. The complete derivation may be found in the extended version of this paper [18]. Our equation for n using W , the *percentage-wasted* metric is:

$$n \geq \left(\frac{Z_{1-\alpha/2}}{\epsilon} \left(W \left(\sqrt{\frac{\sigma_w^2}{\mu_w^2} + \frac{\sigma_r^2}{\mu_r^2} - \frac{2\sigma_w\sigma_r\rho}{\mu_w\mu_r}} \right) \right) \right)^2 \quad (3)$$

With variance σ_w and σ_r and mean μ_w and μ_r

2.2 Sender's Algorithm

Equation 3 allows us to determine n , the sample size needed to archive a $\hat{W}$ with an error of ϵ and a confidence level of $1-\alpha$. One difficulty with this equation is that it requires knowledge about all of receivers. Specifically, we need to know the entire set of receivers' feedback, set R , which is of size N in order to derive our sample size n . This means we would need to collect the entire set of receiver's feedback in order to determine how big of a sample we need. To overcome this limitation we have

- 1) Estimate N , the total number of receivers in the videoconference using techniques given in [17]
- 2) Initiate the process to receive feedback from δ receivers using techniques given in [17].
- 3) Estimate $\hat{n}$, using Equation 3, based on the receivers' feedback obtained in step 2.
- 4) If $\hat{n} > \delta$ then set $\delta = \hat{n}$.
- 5) Utilize feedback to determine the sender's transmission rates.
- 6) Go to step 2.

Fig. 1. Sender's Iterative Algorithm to Determine Transmission Rates

developed an iterative algorithm. This algorithm is shown in Fig. 1. The algorithm is iterative in the sense that we continually estimate n based on a subset of the receivers feedback rates and adjust our sample size according to this newly calculated value. At the same time the sender uses the receivers feedback to calculate its transmission rates.

In order to obtain a subset of the receivers' bottleneck rates our algorithm utilizes a technique developed in [17]. Their approach utilizes probabilistic probing in order to estimate the number of receivers in the videoconference. In addition, they provide an algorithm to obtain a fixed amount of feedback from a randomly determined subset of receivers. We utilize these techniques in steps 1 and 2 in order to obtain the feedback necessary to calculate our new transmission rates.

In step 3 of our algorithm we calculate $\hat{n}$ utilizing equation 3 and the feedback obtained in step 2. For this calculation we are using $\epsilon = .05$ and $\alpha = .05$. It should be noted that this is only an estimate of n since we are utilizing only a subset of the receivers in our calculation. While an averaging of the $\hat{n}$'s might more accurately represent the true n , in order to be conservative in step 4 we only adjust our sample size δ to larger values of $\hat{n}$. In Step 5 of our algorithm we utilize the feedback to determine the sender's transmission rates.

2.3 Real-Time Transport Protocol (RTP)

An alternative to the algorithm presented in Fig. 1 is the one used by RTP. RTP provides for feedback scalability via its RTP Control Protocol (RTCP) [11]. In this approach the amount of feedback for all receivers is limited to a predetermined "session bandwidth". In RTCP feedback packets are multicast to all group members, both senders and receivers. Multicasting is necessary in order for each receiver to be able to determine the current video session size and therefore the amount of bandwidth being used by the RTCP. As the number of receivers grows the receivers increase the amount of time before sending a feedback packet, which decreases the overhead of the feedback packets.

There are two negatives to this approach. First, all links in the network carrying video traffic will experience the overhead of the entire "session bandwidth" amount

of traffic since the feedback is multicast to all receivers. In comparison in the LCMP approach the source controls the amount of feedback required. This allows our algorithm to unicast feedback packets between the receiver and the source. This reduces the overhead of feedback packets on the network. Second, the RTP feedback approach does not take into account the distribution of the bandwidth at the receivers. In the RTCP approach as the number of session grows the amount of time a receiver waits to send its feedback increases. This technique does not take into account the effect of the reduced feedback on the sender's control algorithm. In the LMCP approach we continually calculate the amount of feedback required allowing us to scale the feedback based on the current distribution of the receiver's bandwidths.

3 Performance Analysis

In order to determine the effectiveness of the algorithm given in Fig. 1, we have analyzed its performance with hundreds of distributions. The receiver bottleneck rate distributions varied from a small number of clusters, to very clustered, to uniformly distributed.

There are two key steps in our algorithm. The first is the calculation of $\hat{n}$ in step 3. Due to the correlation between w_i and r_i we have found that a fairly large δ , our minimum sample size, is necessary in order to achieve a realistic initial $\hat{n}$. For our analysis we have initially set $\delta = 120$. Fig. 2 shows histograms of the distribution of the receiver's bottleneck rates used in our first set of simulations. The bottleneck rate represents the slowest rate on the path between the source and each receiver. As this figures show we looked at receiver distributions ranging from clustered to uniform. Fig. 3 shows histograms of the calculated $\hat{n}$ for 60 iterations of the sender's algorithm for the 6 receiver bottleneck rate distributions shown in Fig. 2. The number shown (e.g. true $n=?$) on each histogram is the true value of n calculated using all 500 receivers. As we can see by these histograms, the estimated value $\hat{n}$ tends to be normally distributed around the true n . This is what we would expect and means that equation 3 gives us realistic estimates of n for the given distributions.

In order to understand the overall effectiveness of the algorithm we need to look at its effect on the metric *percentage-used*. Fig. 4 shows the calculated value for the metric *percentage-used* for 50 different receiver bandwidth distributions. This *percentage-used* was based on the transmission rates as determined by our sampling algorithm (Fig. 1). This allows us to see how accurately our algorithm performs. The 50 different receiver bottleneck rate distributions ranged between slightly clustered to uniformly distributed. Fig. 4 shows the variations in the metric *percentage-used* using these 50 distributions. The short lines in this graph represents the difference between the true or ideal *percentage-used* achieved using all N receivers' feedback and the worst-case or minimum *percentage-used* achieved for 60 iterations of our algorithm. As you can see from this graph, our algorithm achieved nearly optimal performance for all distributions with a maximum variation for any of the 50 distributions of .05. In addition, our estimated sample size $\hat{n}$, never exceeded 200 for the 50 distributions and was below 100 for the majority of the distributions.

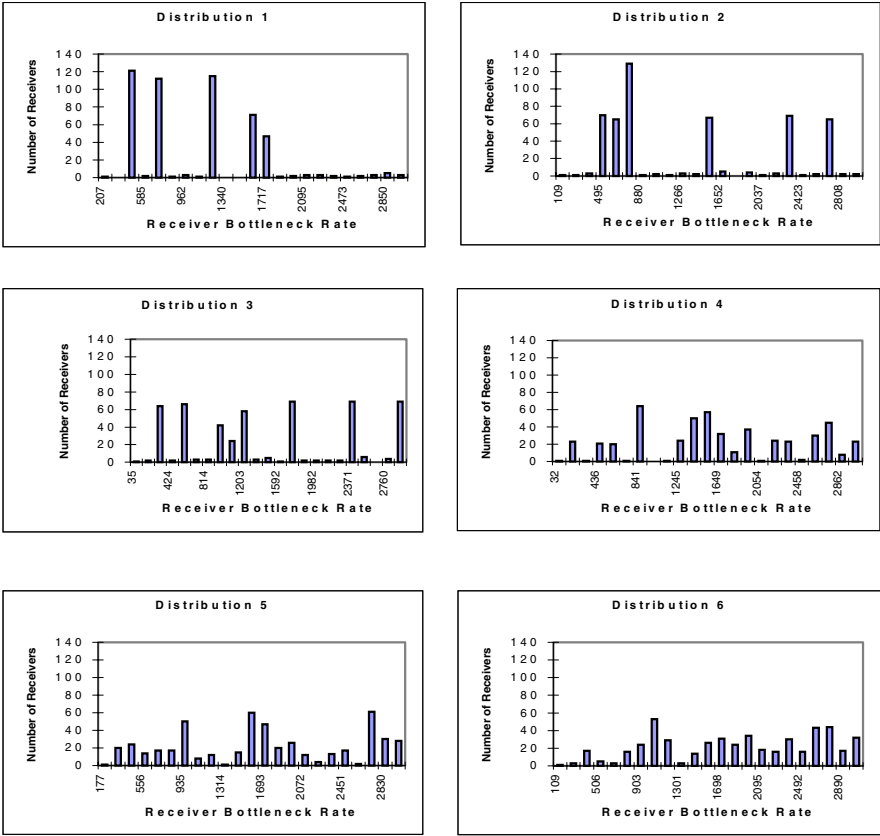


Fig. 2. Histograms of the Receivers' Bottleneck Rate Distributions, with $N = 500$ and bottleneck rates in K bits per seconds (kbps)

4 Summary

In this paper we address the issue of the scalability of the receivers' feedback for large multicast videoconferences. In our previous work we developed the LMCP to control the transmission rate of the video source. One weakness of this approach is that it required feedback from all receivers in order to maximize the receivers' displayable video. In this paper we introduced an algorithm that utilized a sample of the receivers' feedback in determining the sources' transmission rate.

This algorithm first estimates a sample size n such that $P(|P - \hat{P}| > \epsilon) \leq \alpha$, where P is the true *percentage-used* and $\hat{P}$ is calculated based on our sample. The algorithm then samples the receivers' feedback rates and determines the sender's transmission rates. Our analysis showed that this algorithm achieved a $\hat{P}$, which varied at most .05 for the distributions presented. In addition for the 50 distributions presented our maximum feedback rate was 2kbps for 500 receivers

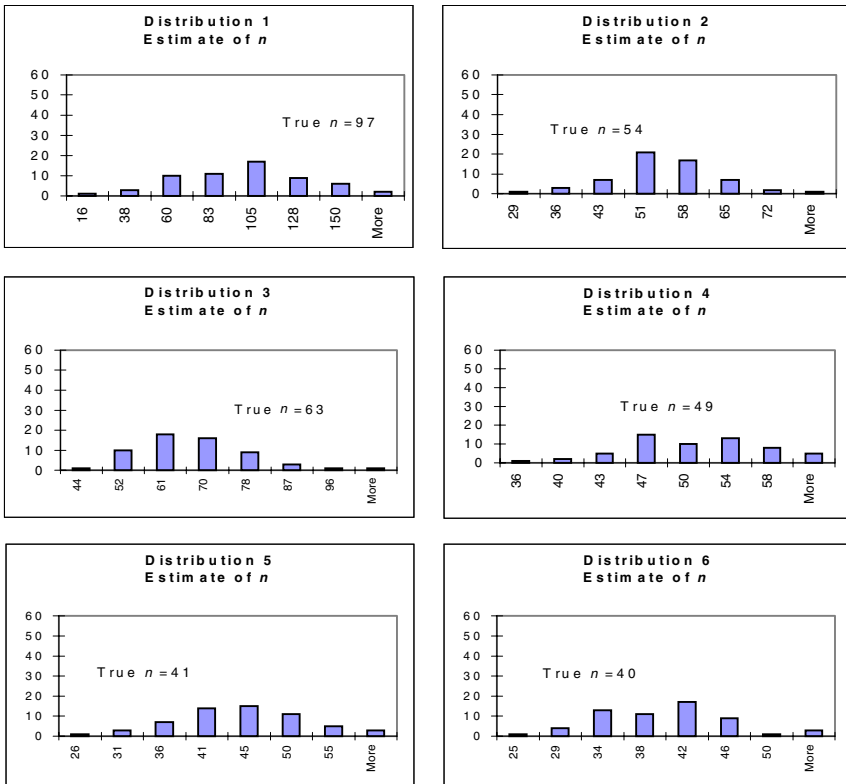


Fig 3. Histograms of the calculation of $\hat{n}$ for the six different receiver bandwidth distributions found in Fig. 2. The calculation for $\hat{n}$ was done with the $N = 500$, $\varepsilon = .05$ and $\alpha = .05$. Each histogram represents 60 iterations of the algorithm.

References

1. Blake, S., et al., *An Architecture for Differentiated Services*, RFC 2475, 1998.
2. Busse, I., B. Duffner and H. Schulzrinne, *Dynamic QoS Control of Multimedia Applications Based on RTP*, First International Workshop on High Speed Networks and Open Distributed Platforms, St. Petersburg, Russia 1995.
3. Smith, H., Mutka, M., Rover, D., *A Feedback Based Rate Control Algorithm for Multicast Videoconferencing*, Journal of High Speed Networks pp. 259-279 1998.
4. Cheung, S., H. Mostofa, M. Ammar and X. Li, *On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution*, INFOCOM, San Francisco, CA 1996.
5. Hoffman, D. and M. Speer, *Hierarchical Video Distribution over Internet Style Networks*, International Conference on Image Processing 1996.
6. McCanne, S., Thesis: *Scalable Compression and Transmission of Internet Multicast Video*, University of California at Berkeley, 1996.
7. McCanne, S., V. Jacobson and M. Vetterli, *Receiver-Driven Layered Multicast*, SIGCOMM, Stanford, CA 1996.

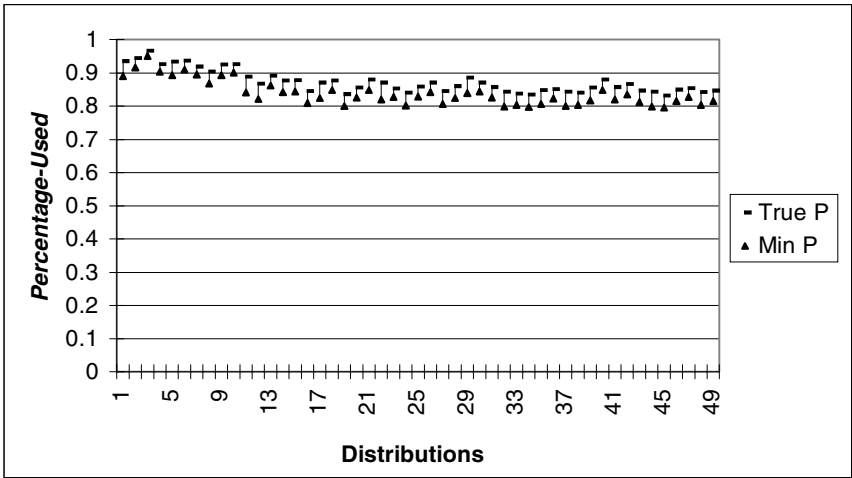


Fig. 4. True *percentage-used* versus minimum *percentage-used* based on our algorithm for 50 different distributions of $N=500$ receivers, $\epsilon = .05$ and $\alpha=.05$. Each line in the graph represents 60 iterations of our algorithm.

8. Vicisano, L., L. Rizo and J. Crowcroft, *TCP-Like Congestion Control for Layered Multicast Data Transfer*, INFOCOM, San Francisco, CA 1998.

9. Smith, H. and M. Mutka, *Fair Link Sharing with Layered Multicast Videoconferencing*, Globecom 2000, San Francisco CA 2000.

10. Smith, H., M. Mutka and E. Torng, *Bandwidth Allocation for Layered Multicast Video*, International Conference on Multimedia and Computer Systems, Florence, Italy 1999.

11. Schulzrinne, H., S. Casner, R. Fredrick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications" *RFC 1889* (Internet Engineering Task Force, RFC1889, 1996).

12. McCanne, S., M. Vetterli and V. Jacobson, *Low-Complexity Video Coding for Receiver-Driven Layered Multicast*, IEEE Journal of Selected Areas in Communications pp. 983-1001 1997.

13. Shapiro, J., *Embedded Image Coding Using Zerotrees of Wavelet Coefficients*, IEEE Transactions on Signal Processing 1993.

14. Fenner, W., *Internet Group Management Protocol, version 2*, Internet Engineering Task Force, Internet Draft RFC2236, 1996.

15. Shacham, N., *Multipoint Communication by Hierarchically Encoded Data*, INFOCOM, Florence, Italy 1992.

16. Conover, W., *Practical Nonparametric Statistics* (John Wiley & Son, Inc., 1998).

17. Bolot, J., T. Turletti and I. Wakerman, *Scaleable Feedback Control for Multicast Video Distribution in the Internet*, SIGCOMM, University College London, London, U.K. 1994.

18. Smith, H., Mutka, M., Yang, L., *Feedback Scalability for Multicast Videoconferencing, Extended Version*, www.csc.calpoly.edu/~husmith

Group Communication and Multicast

Justin Templemore-Finlayson and Stan Budkowski

Institut National des Télécommunications,
Département Logiciels-Réseaux,
9 rue Charles Fourier,
91011 Evry Cedex, France

Abstract. In this paper we propose a group communication architecture for the Internet. This architecture is implemented as a middleware which provides a separation between a general *group communication* service model, and the *multicast* mechanisms used to implement this service. This middleware provides a web-oriented framework for implementing end-to-end group communication services as constrained by specific application needs and available network technologies.

1 Introduction

Multicast is a means of *one-to-many* communication. It is essential for scalable group communication as it allows a group member to communicate with once with an abstract group and yet effectively reach multiple other members of the group. The main objective of multicast is to eliminate wasteful duplicates in the network when sending the same information to multiple receivers.

Group applications today operate without multicast by using a repeated unicast scheme, in which a data sender opens a separate unicast connection to each data receiver. This is expensive to both the sender and the network, imposing high a bandwidth-delay product on transmissions at the sender and creating as many duplicates of the data sent as there are receivers in the group.

In the Internet context, multicast research has focused around IP multicast [4,5], a network-layer solution for distributing data from a sender to multiple receivers. IP multicast ensures that a packet sent to a multicast group will only traverse a given physical link in the network at most once, independent of the number of receivers of that packet.

IP multicast has met with only moderate success however, due in part to the slow pace of network-level deployment, but more importantly due to difficulties related to protocol standardization and concerns about network security in the presence of a widely available IP multicast service.

Recent research efforts including [10,1,2] have proposed a hybrid multicast approach, coined *Application Layer Multicast* (ALM) which performs multicast routing at the application layer. ALM approaches aim to perform multicast

distribution between endsystems involved in a group communication using only unicast network primitives.

While the efficiency of ALM protocols is acknowledged as being less than that of IP multicast [2], this approach is promising as it approaches IP multicast efficiency without the difficulties or dangers of infrastructural deployment.

One area in which ALM is being successfully applied is in the new field of *content distribution* companies. Relay servers are placed at strategic points in the Internet, closer to a clustering of customers than the originating server. “Content” (such as streamed video and audio, or proxy data) is relayed by unicast to the server, to which clients also connect using unicast. The content distribution company is thereby able to reach a much larger number of clients at reduced cost to both the company and the network than if each client was being served directly from the originating server.

Multiple relay servers can be connected using unicast tunnels to provide a virtual multicast network over the Internet, for which reason ALM is also known as *overlay multicast*. The MBone [8] is one example of an overlay network.

Many group applications exist today which are forced to use a repeated unicast scheme and would benefit greatly from multicast distribution. Common examples that we use daily include real-time multimedia streaming, multi-player games or forums and audio-visual conferencing/teaching software.

These diverse applications all use a common *group communication* service, but have very different and often conflicting requirements of this service. Consider for example a multimedia streaming application which is highly delay sensitive but which can permit loss, compared to a file-sharing application which tolerates some delay but no loss. No single protocol implementing the service could meet these diametrically opposed requirements.

The Internet Multicast Architecture (IMA) proposes an abstract service model for *group communication* in the Internet. It is conceived as a middleware layer which allows protocols to implement the service as constrained by an application’s service requirements and the available network services.

This allows the development of protocol instances in a consistent group communication framework. It enables a protocol to fulfill an application’s specialised group communication requirements, and to provide this same service in a transparent way using IP multicast or Application Layer Multicast.

The rest of this paper describes the IMA, and the separation of *group communication* and *multicast*. Section 2 presents the IMA group communication service model for the Internet. Section 3 presents a middleware which implements this service. Section 4 describes concurrent ALM projects which propose an Internet group architecture. Section 5 concludes with some interesting new possibilities created by Application Layer Multicast, and which we are pursuing in the framework of the Internet Multicast Architecture.

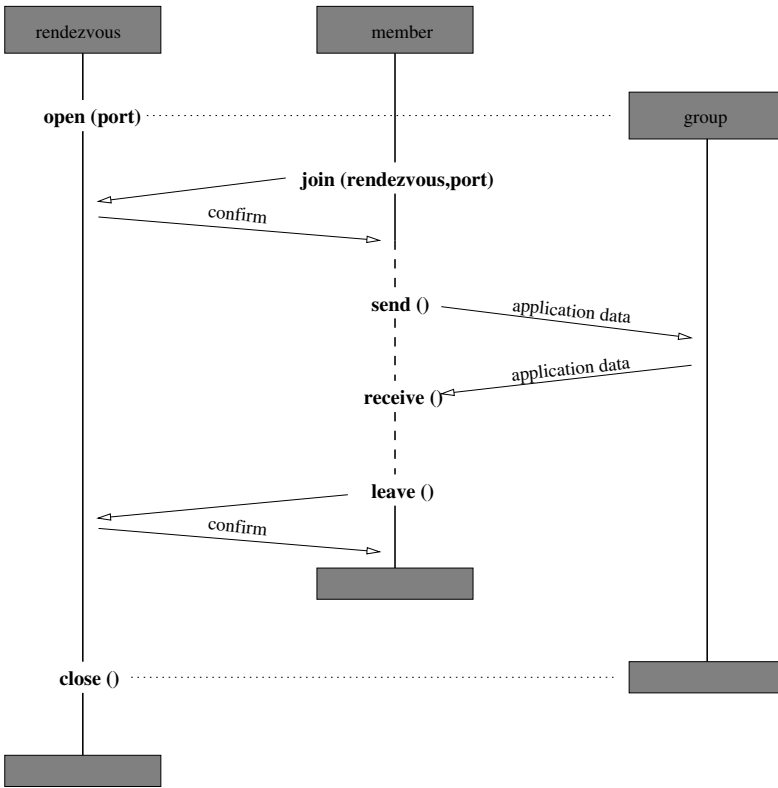


Fig. 1. A service model for group communication

2 A Group Communication Service Model

We define *group communication* as the communication that takes place between an individual and some abstract group representing other participants in that communication. The implication is that application data sent by one participant to the abstract group is distributed to all the other participants.¹

In the following we define an abstract service model for group communication in the Internet. This model describes the IMA framework for group creation, group addressing and discovery, and group communication. This service model is independent of both the multicast technique(s) used to distribute application data between members of the group and the service. These are defined in *protocol instances* which implement the group model. Section 3 describes some protocol instances developed for the IMA and the services they provide.

¹ We do not wish to say that all group communication is reliable, as some services like IP multicast are best-effort. We mean that some attempt is made to deliver data sent by one participant to all the others.

The group communication service model defines three group communication entities:

1. A **rendezvous** or meeting point for group members. The rendezvous provides a central point for group management. The rendezvous's functions include group creation and tear down and group access control. The rendezvous is not however implicated in data distribution between the members of the group.
2. A **member**. A group will have many members. A member is a participant which can send to, and/or receive from, a group. A member joins a group by addressing itself to the rendezvous. If permitted to join the group, it is provided with a protocol specific information which allows it to communicate directly with the group.
3. A **group**. The group is an abstraction of the all other members which have joined to the rendezvous. Once a rendezvous has created a group, and a member has joined it, the member communicates with the other members by way of the abstract group. The specifics of communication with a group are left to individual protocol instances.

The interaction of these entities is shown in Figure 1.

In Figure 1, a rendezvous host *opens* a group on a local port. The address of the rendezvous and the port number define a globally unique address for the group.

A group member *joins* a group by connecting to a known group address. Once accepted, it may exchange datagrams with the abstract group. The actual distribution of this application data from a member to the group depends on the actual protocol instance implementation.

A group member *leaves* the group by re-contacting the rendezvous. Only the rendezvous may *close* the group.

3 Group Communication Middleware

The IMA implements the group communication service model of Section 2 as a middleware layer, shown in Figure 2. This middleware provides a homogeneous group communication service over different available network technologies. This is useful in the current Internet situation where group applications are used globally but IP multicast is being deployed and activated incrementally.

The IMA middleware defines a set of abstract group communication primitives which together make up what we call a *GroupSocket*. A GroupSocket provides the group communication interface between a group member and the rest of the group. A Java definition of an abstract *Group Socket* object is listed in Figure 3.

An IMA *protocol instance* implements this abstract class. An implementation is based on the available network technology (Application Layer Multicast, IP multicast, or a combination), and the service requirements of a class of group applications. The service requirements also define the roles of different group

members, and partitions the availability of the *GroupSocket* primitives accordingly. We are currently investigating four protocol instances with different service requirements and network constraints:

1. **Reliable Streaming** : A reliable, one-to-many bulk data transfer service, similar to that of the Reliable Multicast Transfer Protocol II [9]. This protocol instance uses an overlay tree of TCP connections to transmit files reliably from a single sender to multiple receivers. Its defines two roles: A *sender-rendezvous* which creates and sends to a group, and a *receiver* which joins and passively receives data from a group.
2. **Reliable Messaging** : A reliable, many-to-many communication service, similar to that of the Scalable Reliable Multicast protocol [7]. This protocol instance uses an overlay ring structure of TCP connections to provide partially-ordered communication between its members. It defines two roles: A *rendezvous-server* which manages the group, and a *client* which can both send and receive packets reliably to the group.
3. **Basic Service over IP multicast** : An unreliable datagram single-source multicast service. This protocol instance is a lightweight wrapper to IP multicast. A *source* controls access to the group and acts as a sender, and a *receiver* passively receives.
4. **Basic Service over UDP**: Another unreliable datagram single-source multicast service. This protocol instance however provides the basic service using an Application Layer Multicast tree structure and simple (unicast) UDP services.

The IMA approach enables us to break out of the shortest path tree approach imposed by IP multicast. It encompasses ALM and overlay network approaches, which are able to use alternative distribution methods, hopefully better suited to the targeted application's communication requirements. For example, we are investigating a ring topology for small conferencing applications (see the **Reliable Messaging** protocol instance above), which promises better group performance than using an IP multicast tree for each participating member.

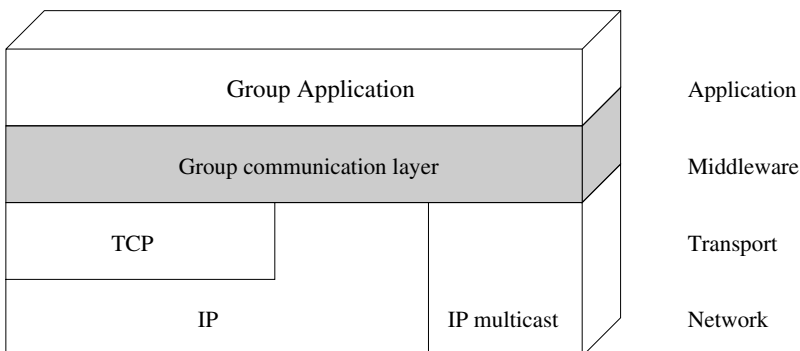


Fig. 2. The group communication service as middleware

```

abstract class GroupSocket {

    // Create an IMA Group on rdvPort. The IMA Group Address
    // of this IMA group will be (localhost:rdvPort).
    abstract void open (int rdvPort);

    // To this IMA group.
    abstract void close ();

    // To join the parameterized IMA Group
    abstract void join (InetAddress rdvAddress, int rdvPort);

    // Leave the currently joined IMA Group.
    abstract void leave ();

    // Receive an ADU from the group
    abstract GroupDatagram receive ();

    // Send an ADU to the group
    abstract void send (GroupDatagram dgram);

}

```

Fig. 3. Abstract definition of a GroupSocket as a Java class

Another important advantage offered by the IMA architecture is the ability to implement the same service using different available distribution technologies, as seen by the two variants of the **Basic Service** protocol, one of which uses IP multicast, the other Application Layer Multicast. This enables a group application to be provided in all environments, and facilitates a smooth evolution process to emerging technologies.

4 Related Work

Reliable Multicast ProXies (RMX) [1] splits a large heterogeneous multicast group into a number of co-located and homogeneous data groups, each with its own RMX. Data sent to the group is distributed between RMX's using TCP, and within a data group using Scalable Reliable Multicast (SRM) [7]. The principal of Application Level Framing is followed, allowing insertion of application semantics in an RMX. An application can specialize an RMX in terms of data reliability, transmission scheduling and dynamic data transformation.

RMX defines an overlay network architecture which requires the placement of third-party servers in the network, while the IMA is designed to be able to provide group communication without any network support, even between end-systems.

Nevertheless, the IMA is able to support a two-tier approach as proposed by RMX. We propose inter-domain unicast tunnels with an IP multicast-based protocol in local subnets as a solution to inter-ISP multicast traffic. This is described in [10].

Narada [2] presents an excellent case for Application Layer Multicast, which they refer to as *end-system multicast*. The arguments presented in this paper are perfectly valid in the context of our work on the IMA. [2] does not present a multicast architecture, but is rather an investigation into the performance of Application Layer Multicast compared to IP multicast protocols.

In relation to the IMA, Narada presents a protocol instance providing a many-to-many reliable service. Narada implements this service by creating a mesh overlay structure between members and running the IP multicast Distance Vector Multicast Routing Protocol [6] to create a multicast distribution trees for each sender in the group.

5 Conclusions

Research into group communication has for a long time been neglected in favor of multicast technology. Now that the advantages of both network and application-layer multicast approaches have become recognized, a renewed look at group communication - independent of multicast technology - is required.

In this paper we define group communication as an abstract service which can be separated from the multicast technology used to implement it. We wish to make it clear that multicast is simply a supporting technology, and that the group application service can in fact be provided without this technology.

In Section 2 we propose a group communication service model for the Internet. This service definition hides the multicast (or unicast) technology used to implement a group communication service, which means that group applications can evolve to new technologies (such as IP multicast) without requiring application re-engineering. For the same reason, it makes it possible to support a homogeneous group communication service irrespective of the underlying network.

In Section 3 we present the Internet Multicast Architecture, a unifying architecture which implements the group model of Section 2 as a middleware group communications layer. The main element of this architecture is the **Group Socket** class, which defines a standard interface to group communication protocols whether they distribute data using “traditional” IP multicast, or an Application Layer Multicast approach.

The IMA provides the opportunity to develop protocol instances as a function of the group application needs and the available distribution technology (multicast or unicast). This enables protocols to be developed to best meet the application needs, which is not possible when only IP multicast is used. The IMA also requires no network deployment, and can at its simplest operate directly between endsystems.

6 Future Directions

The Application Layer Multicast approach opens some other interesting avenues of research which we are exploring.

Experience has shown that the separation of multicast distribution and end-to-end services at the network-endsystem boundary makes the provision of group-wide services such as reliability and flow control difficult. ALM on the other hand allows the close integration of end-to-end services with the multicast distribution. We are leveraging this close association to develop a working reliable multicast protocol instance for general use in the Internet, an as yet unresolved academic problem.

Another interest aspect of Application Layer Multicast is the possibility to use *adaptive routing* of data between members of a group. This promises better congestion avoidance and network load balancing in reliable multicast groups. A future paper [11] presents some promising results for this approach.

Application Layer Multicast also provides us with the freedom to use other distribution structures than the multicast tree imposed by IP multicast. [3] suggests that a ring structure can be suitable to reliable communications in some situations than a tree. We are part of an international cooperation developing Internet group applications and to investigate different distribution structures for group communications.

References

1. Y Chawathe, S McCanne, and E Brewer. RMX: Reliable multicast for heterogeneous networks. In *Proceedings of IEEE INFOCOM 2000*, March 2000.
2. Y-H Chu, SG Rao, and H Zhang. A case for end system multicast. In *Proceedings of ACM Sigmetrics*, June 2000.
3. S Chen, B Yener, and Y Ofek. Performance trade-offs in reliable group multicast protocols. In *Proceedings of the IEEE INFOCOM '99*, pages 982 – 989, March 1999.
4. S Deering. Multicast routing in internetworks and extended LANs. In *Proceedings of the AGM SIGCOMM 88*, pages 55 – 64, August 1988.
5. S Deering. Host extensions for IP multicasting. RFC 1112, August 1989.
6. S Deering, C Partridge, and D Waitzman. Distance vector multicast routing protocol. RFC 1075, November 1988.
7. S Floyd, V Jacobson, S McCanne, C Liu, and L Zhang. A reliable multicast framework for light-weight sessions and application-level framing. *Computer Communication Review*, 25(4), August 1995.
8. IETF MBONE deployment working group. <http://www.ietf.org/html.charters/mboned-charter.html>.
9. Sanjoy Paul, Krishan K Sabnani, John C Lin, and Supratik Bhattacharyya. A reliable multicast transport protocol (RMTP). To appear in the IEEE Journal on Selected Areas in Communications, special issue on Network Support for Multipoint Communication.
10. J Templemore-Finlayson and S Budkowski. IMA: An internet multicast architecture. Research Report RR 00004-LOR, Institut National des Télécommunications, Evry, July 2000.
11. J Templemore-Finlayson and S Budkowski. An overlay approach to reliable multicast. In *Proceedings of AFRIKOM 2001*, May 2001.

Avoiding Counting to Infinity in Distance Vector Routing

A. Schmid, O. Kandel, and Ch. Steigner

{axs, olaf.kandel, steigner}@uni-koblenz.de

Abstract. The Routing Information Protocol (RIP) may occasionally introduce misleading routing information into the routing table, due to network topology changes such as link or router failures. This is known as the “counting to infinity” problem. In the past, the distance metric had to be below 16 hops, in order to keep this counting within reasonable limits. In this paper a more elaborate approach is presented in order to recognize those router interfaces, which might have received misleading routing messages. This is accomplished by evaluating routing updates more carefully than is done by the well known split horizon approach. In contrast to other approaches, the router interfaces are examined in pairs to determine if a loop exists between them. The algorithm locally extracts all the information it needs from the normal update messages that are exchanged between RIP neighbors and is thus executed in constant time. Only some minor calculations have to be carried out to gain the knowledge that is necessary to recognize those interfaces which may have received misleading routing information. Hence, this distance vector routing without “counting to infinity” can be used in complex networking environments.

Keywords: Routing Information Protocol, Internet, Protocol Design, Counting to Infinity

1 Introduction

The major advantages of distance vector routing (DVR) are minimal exchanges of routing messages, minimal administration, and minimal memory and processing requirements. DVR is therefore still widely used in spite of its known weakness. Namely, that it may suffer severely from network component failures due to the “counting to infinity” approach.

This paper analyzes the preconditions of routing loops. It examines the situations which are prone to routing loops, shows how to avoid routing loops and proves the acquired solution. Simulation results are presented to give a comparative insight into the behavior of conventional RIP and the newly developed Routing Information Protocol with Minimal Topology Information (RIP-MTI).

2 Distance Vector Routing

DVR protocols exchange very little information among neighbor routers. These “route advertisements” consist of destination and metric, where the metric is an integer number corresponding to the actual or prescribed distance to the specified destination network.

In this paper only one destination network, called d , is regarded at a time. This convention is used throughout (see fig. 1).

Routing is understood as the process of providing routing information for routing tables, whereas forwarding is the delivering of packets by using the routing table information. In most DVR protocols the routing table contains the cost of the shortest path and the next router address in the path to the destination network. Routers directly connected to destination d know the link cost, and send this information to their neighbors. Other routers compute the shortest path to d as the minimum of the sum of the information received from their neighbors and the link costs to these neighbors.

Within RIP, all routers know the destination addresses to their neighboring networks and send them to their neighboring routers with a metric of 1. These routers themselves send the received destination address with metric 2 to their neighboring routers. This process is repeated until all routers within the autonomous system (AS, region of the Internet under the administrative control of a single entity) know all destination addresses and are provided with reachability information for all destinations within the AS.

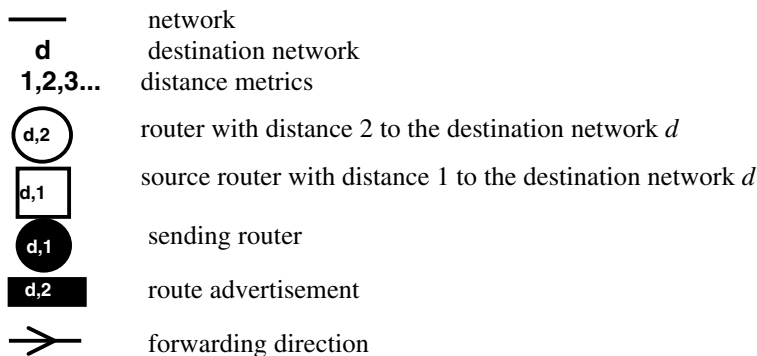


Fig. 1. Legend for the network/router diagrams

In order to calculate the shortest link to the destination network, only the smallest metric is stored in the routing tables. Routers send routing updates periodically (every 30 seconds) reporting all the routes learned from their neighbor routers. If a route is not advertised for three minutes or more, this route is marked as “unreachable” by assigning the “infinity” metric (16) to it. If, however, other routers are still advertising a route to this destination, the best among them is selected. Fig. 2 shows an example of how routers exchange routing updates.

What are the disadvantages of RIP? RIP is able to produce the right routing table entries: it reacts quickly to good (i.e. shorter routes) but incorrect news, and very

slowly to bad (i.e. longer routes) but correct news [13]. The main problem is that “routing loops” can arise with reachability information circling inside them (fig. 3).

How do routing loops arise? As soon as a router recognizes that a certain destination network is unreachable, it propagates this information throughout the system via neighbor to neighbor propagation. During this period, one of the routers inside of a loop, which has not yet received the “network unreachable” information, sends a wrong but better than infinity update message to its neighboring routers. Because this pretended “good news” is considered more valuable than “bad news” all routers will accept and propagate this “pseudo good news”, which may finally lead to counting to infinity if this “good” news is propagated further inside the routing loop.

The lifetime of these misleading information in routing tables can be limited if a low maximum value for the longest path in an autonomous system is given [2]. RIP thus only allows 15 routers within one route leading to any destination. If the number of routers on a certain path exceeds 15, the advertised destination is considered to be unreachable.

In fig. 3, three routers i , $r1$ and $r2$ are connecting four networks. Fig. 2 shows the same system, however in a state where no counting to infinity has occurred, due to the fact that the reachability to all networks is given at a distance below “infinity”. In fig. 3, it is shown what happens if network d suddenly becomes unreachable. Unfortunately, the routers’ reactions contradict one another. Router i sends this “bad but right” news to the other routers, $r2$ advertises that a route to network d still exists with a metric of 2 which is a “good but wrong” news. Router $r1$ accepts this route to destination d , because its metric (2) is smaller than “infinity” and assumes that this route really exists. Router $r1$ assigns 3 (2+1) as the cost and $r2$ as the direction to network d . Router $r1$ sends this information to router i in fig. 3 (c).

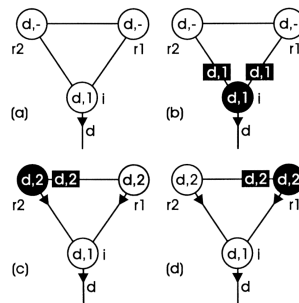


Fig. 2. Building up complete routing information for destination d (“cold start”)

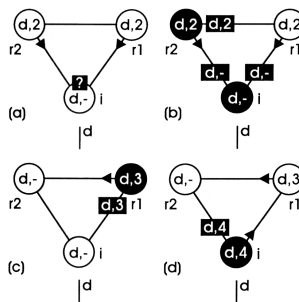


Fig. 3. Routing loop after destination d becomes unreachable

Router i wrongly assumes that the only possible route to reach the destination d is via $r1$. Router i again wrongly advertises the new route to router $r2$ in fig. 3 (d). The metric for destination d in the routing table of router $r2$ is seen as not up-to-date, therefore, router $r2$ wrongly corrects the metric for this route to 5. This procedure goes on within this routing loop until the metric reaches “infinity” in all routing tables, which makes RIP very inefficient in this case. By applying some heuristics, which will be explained below, this inefficiency of RIP can be overcome to a certain degree.

2.1 Improving Convergence

To improve the convergence of RIP, four concepts have been suggested:

- triggered updates
- split horizon
- poison reverse
- path hold-down

These concepts have been integrated into different RIP implementations, but they do not solve the “counting to infinity” problem completely.

Triggered updates are sent immediately after any change of a routing table entry. This was proposed to spread changes of topology faster than before. Routers do not have to wait until the next regular update. The main disadvantage of triggered update is that the routing traffic rises strongly after a failure of a network component. To reduce this effect, routers are forced to wait a random length of time before they send a triggered update.

Split horizon avoids some of the incorrect routing advertisements:

“A router does not send outgoing route advertisements back to the router from which they were learned.”

This approach reduces the probability of routing loops. It does not, however, avoid them. In any routing loop, there are destination networks, which are advertised with the same metric via two or more interfaces of a router. If a router decides to accept one of its neighbor’s routes as the best route, and sends this wrong but perceived good routing update to other routers, a routing loop may be created.

Path hold-down establishes a period of time (typically 60 sec.) during which a router will ignore new routing information about a given network, once the router has learned that this network is unreachable, i.e. has been assigned the metric “infinity”. According to RFC 1009 [1, p24], a hold-down period is chosen long enough to allow for the unreachable status to propagate to all routers in the autonomous system (AS). The avoidance of additional network load is considered more important than fast route re-establishment in case of a failure. Therefore, this is the slowest approach in the group, but it should avoid most routing loops. The earlier releases of IGRP used a path hold-down technique, the latest releases use poison reverse only [8, p 117].

2.2 Other Approaches

One way of avoiding the counting to infinity problem is to recognize routing loops by gaining some knowledge about the path back to the source router. Cheng, et. al. [2] and Rajapopalan and Faiman [7] suggested appending the router nearest to the destination (which is called the head of path) to the routing update. With this information, it is possible to trace back the path from the destination to the source router. This back-tracing, which needs additional exchange of information between routers, allows the recognition of routing loops within the path.

Including the router-label nearest to the neighbor router in the update message in order to obtain a three node path knowledge was proposed by Shin and Chen [12] to avoid two-node looping. This algorithm can be extended to a k -th order algorithm which avoids all loops with more than k hops. However, this leads to an increase in the size of update messages and the local memory requirement increases in proportion to k . A DVR protocol developed for routing between AS-domains, called the Border Gateway Protocol (BGP), specifies the entire path from source to destination in the update message.

Due to the fact that a routing loop needs certain timing conditions to develop, routing loops can be avoided by coordinating the exchange of routing information. Jaffe and Moss [5] showed that no routing loops can occur in DVR algorithms after a link addition or a link-cost decrease. Their protocol requires inter-nodal coordination if link costs increase or resources fail in the network. Garcia-Luna-Aceves proposed the Diffusing Update Algorithm (DUAL) [3,4], where the routers coordinate themselves mutually by confirmation messages. With these messages, a router can find out if the routing update procedure it has initiated to all its neighbors has terminated already. If not, the router is blocked: it accepts no further update messages, thereby avoiding the creation of routing loops. The implementation of this method is extensive. It cannot be carried out as an add-on to RIP. According to Garcia-Luna-Aceves [2,4] there are three feasibility conditions, whereby one of each guarantees freedom of routing loops (Distance Increase Condition DIC, Current Successor condition CSS, and Source router Node Condition SNC). It should be noted that these conditions guarantee the absence of routing loops, however, they are not able to calculate shortest paths. DUAL is used by Cisco, a router hardware vendor, under the name EIGRP (Extended Interior Gateway Routing Protocol). A coordinated diffusing computation is initiated only when distances increase.

In summary, it can be said that the alienation to RIP in the Internet community can be traced back to the unsolved problems with “counting to infinity” and the resulting slow convergence of RIP. All the aforementioned solutions require the extension of routing messages in order to function properly.

3 The RIP-MTI Approach¹

In order to avoid routing loops locally without any change of RIP messages, RIP's behavior of forgetting all but the shortest path is changed. “Minimal Topology Information” (MTI) can be derived from the information provided by the other paths.

¹ see [9,10,11]

The RIP-MTI approach is based on two simple ideas:

- router-local recognition of **routing loops** between all interfaces of single routers
- router-local recognition of **source loops** by the conditions given later in the theorems 4, 5 and 6

Source loops are loops which pass through two **source router interfaces** more than once on their way through the networks and routers (see fig. 5).

In two local tables on each router, the existence of loops between a pair of interfaces and the network environment are stored. A loop is recognized by examining reachability information for the same destination network, found on two different router interfaces. Thus, a loop can be recognized by two routing updates which contain the same destination network and arrive over different interfaces of the same router. Important: the loop information stored consists of the beginning and the ending router interface and the distance metric for the loop.

This additional loop information is exactly what is needed to be able to decide whether a destination network d can be reached through more than one interface. This is the fundamental idea of MTI.

The RIP-MTI algorithm also makes use of the other concepts for suppressing “counting to infinity”. RIP-MTI routers must use split horizon and triggered updates, may use poison reverse, and of course must use MTI, which can be seen as a very fast **conditional path hold down** for wrong alternative routing updates. The further explanation is based on link costs of 1 - it will be shown in part III. E how a link metric with arbitrary costs can be used as well.

Loops may have two meanings for RIP-MTI. **First**, they may be used to offer alternative reachabilities to a destination network if they do not pass any router more than once and one path to the destination network fails, which is referred to as “cycle” later on. **Second**, they may be memories for wrong reachability information if they pass a router more than once. This is referred to as “source loop” later on.

Fig. 4 depicts the behavior of router i knowing that there has been no cycle between interface A (connects destination network d) and interface C (connects network which contains router $r1$). Router i rejects the wrong alternative routing update from interface C because it detects a source loop.

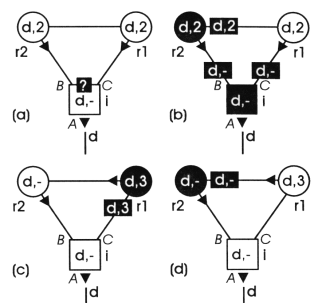


Fig. 4. Routing loop avoided after destination d becomes unreachable.

3.1 Definitions

Definition 1 (Path): A path from router $i = r_1$ to subnet $d = s_L$ is abbreviated as $P^{i,d}$ and is a sequence of routers and subnets:

$$P^{i,d} = (r_1 = i, s_1, r_2, s_2, \dots, r_L, s_L = d)$$

L is the total length of the path. A path from i to d over interface A is abbreviated as $P_A^{i,d}$ and a path that returns back to router i over interface B as $P_{A,B}^{i,d,i}$.

Definition 2 (Route): A route $m_A^{i,d}$ from router i to subnet d over interface A is defined as $m_A^{i,d} = L$, when a path $P_A^{i,d} = (r_1 = i, s_1 = A, r_2, \dots, s_L = d)$ exists.

A routing update $\langle m_A^{i,d} = 1 \rangle$ from router i received by router r_2 over interface A is a route $m_A^{i,d} = m_A^{r_2,d} + 1$.

It should be pointed out, that a path/route is never static. A path/route cannot be assembled out of the routing information stored in the routing tables of the routers at any particular point in time. Time has a very important role, because it always takes some amount of time for a data packet to traverse a given path: during this time, the routing table entries can change because of routing updates the routers may have received in the meantime. The result is, that the path the message traverses, can contain loops, but this is not the case when looking at the routing tables at any single point in time.

A *composite back-route* between the interfaces A and B of a router via destination d is a necessary criterion for a loop between interfaces A and B . In an acyclic network topology there are no composite back-routes.

Definition 3 (Composite back-route): A composite back-route $m_{A,B}^{i,d,i}$ between two interfaces A and B of router i via a destination d exists, if the destination d is advertised as being reachable via interface A and via interface B . Then these two paths can be combined into one composite path

$$P_{A,B}^{i,d,i} = (r_1 = i, A, \dots, d, \dots, s_L = B, r_{L+1} = i)$$

Because a composite back-route is composed of two routes $m_A^{i,d}$ and $m_B^{i,d}$, the metric $m_{A,B}^{i,d,i}$ of a composite back-route can be calculated as follows:

$$m_{A,B}^{i,d,i} = m_A^{i,d} + m_B^{i,d} - 1 = L.$$

A *routing loop* is a necessary condition for the “counting to infinity” behavior.

Definition 4 (Routing loop): A routing loop is a special route $m_A^{i,d}$ (or a special composite back-route $m_{A,B}^{i,d,i}$) which passes through any router only once in its corresponding path.

The *source loop* is a key concept of this work.

Definition 5 (Source loop): A route $m_A^{i,d}$ (or a composite back-route $m_{A,B}^{i,d,i}$) is a **source loop** if it passes the source router i (which receives the route advertisements) more than once. (In fig. 4c router i recognizes that $m_{C,A}^{i,d,i}$ is a source loop)

The following definition of a *cycle* is temporary in the sense that it specifies a cycle but can be reduced later to the minimal cycle.

Definition 6 (Cycle): A cycle $cyc_{A,B}^{i,d,i}$ between two interfaces A and B via the destination d is a special composite back-route $m_{A,B}^{i,d,i}$ between A and B via the destination d , whereby the source router must not be traversed along the route. (In fig. 4 there is a cycle between interface B and C .)

It is possible to define the "smallest metric" of a loop independent of the destination d as a representative of all loops between a pair of interfaces:

Definition 7 (Minimal cycle): The minimal cycle between two interfaces A and B is

$$mincyc_{A,B}^i = \min\{cyc_{A,B}^{i,d,i} \text{ for all destinations } d\}$$

The notion of the *minimal return route* via an interface A is important for further explanations. It is also part of the sufficient cycle criterion (refer to the corollary theorem 6).

Definition 8 (Minimal return route): The minimal return route via an interface A is

$$minm_A^i = \min\{mincyc_{A,S}^i \text{ for all interfaces } S \text{ of } i\}$$

It is crucial for this definition that cycles are the base for minimal return routes, not composite back-routes. Imagine the minimal return route to be the shortest possible route of a conventional message leaving the source router i via the interface A and returning after $minm_A$ forwardings to the source router i .

3.2 Avoiding Source Loops

The existence of two distinct routes to the same destination - a composite back-route - is a necessary, but not sufficient criterion for a cycle.

Therefore all composite back-routes with a source loop are shown in one table (fig. 5). The source loop corresponds to a path $P_{A,B}^{i,d,i}$ which passes through the source router i two times: it leaves router i via interface A , returns to interface A for the first time, leaves i again and finally enters i for a second time via interface B .

A source loop can enter and leave the router via nine different network combinations:

The corresponding path must enter router i again via interface A , interface B or any other interface $S \neq A$ or B and must leave router i via interface A , interface B or any other interface $S \neq A$ or B .

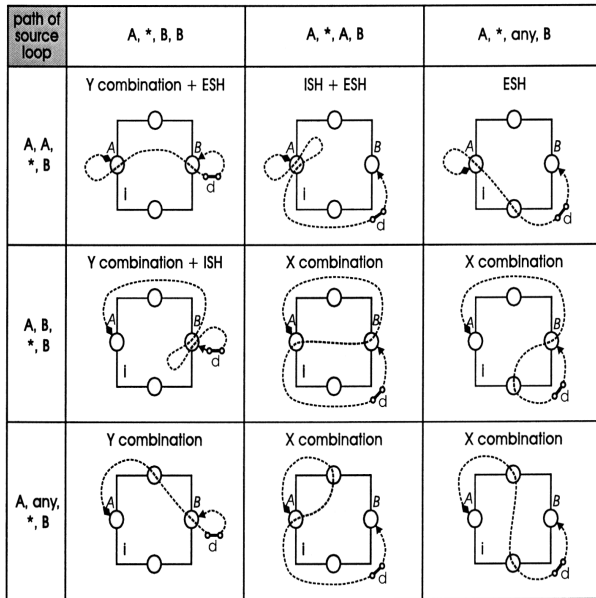


Fig. 5. Enumeration of all possible source loops within a composite back-route.

Fig. 5 provides a complete listing of all source loops using a “rubber band” tightly connected to the interfaces A and B , entering and leaving the router using all possible interface combinations. In fig. 5, the route $m_A^{i,d}$ was chosen to contain the source loop.

For all nine cases the path leaves router i via interface A first and finally enters router i via interface B at the end. Vertically, the interface where the path reenters router i for the first time is varied, i.e. interface A , B or any other interface other than A or B . Horizontally, the interface where the path leaves router i for the second time is varied, i.e. interface A , B or any other interface other than A or B .

The source loops can be differentiated by the following features of their paths:

- **ESH (External Split Horizon):** The path leaves router i for the first time and returns to it via the same interface.
- **ISH (Internal Split Horizon):** The path enters router i for the first time and leaves it for the second time via the same interface.
- **Y combination:** The path leaves router i for the second time and enters it in the end via the same interface.
- **X combination:** The interfaces via which the path leaves router i and returns to it for the **first** time are different, and the same for the interfaces via which the path leaves and returns to router i for the **second** time.

This differentiation has been applied to fig. 5. ESH and ISH can be avoided by applying the split horizon rule, which is proved later on.

Theorem 1

If source loops are avoided by **every** router **locally**, routing loops do not occur **globally**.

Proof by contradiction: let the Path $P^{i,d}$ contain a routing loop. Routing loops contain at least one router r_x which is traversed twice:

$$P^{i,d} = (r_1 = i, s_1, \dots, r_x, s_x, \dots, r_y = r_x, \dots, r_l, s_l = d)$$

From the viewpoint of router r_x , the second traversal of r_x in this path is a source loop.

$$P^{r_x,d} = (r_x, s_x, \dots, r_y = r_x, s_y, \dots, r_l, s_l = d)$$

With router r_x avoiding this source loop, the source loop does not exist in its routing tables and is not propagated to other routers $r_{x'}$. The result is that path $P^{i,d}$ cannot be a routing loop in which router r_x is passed more than once.

In order to show the interrelation between source loops (SL), routing loops (L) and counting to infinity (C) the following conditions hold:

$$\begin{aligned} \neg \exists SL \Rightarrow \neg \exists L & \quad \Rightarrow \quad \neg \exists SL \Rightarrow \neg \exists C \\ \neg \exists L \Rightarrow \neg \exists C & \end{aligned}$$

The first condition is stated by theorem 1, which has been proved already. The second condition can be proved by the following observation: Counting to infinity behavior (C) can be defined as: “C is initiated, if reachability information of a destination subnetwork d runs repeatedly inside a cycle until the advertised distance d reaches the predefined maximum value.” The route to d contains a loop when it enters the cycle for the second time. Hence, if routing loops are avoided, there is no counting to infinity behavior.

Impossible Composite Back Routes

If the split horizon rule is used by the RIP algorithm it can be shown, that five out of nine source loops do not have to be considered, because they do not happen: the entries labeled by “ISH” (internal split horizon) and “ESH” (external split horizon) in fig. 5.

Theorem 2 (Internal Split Horizon, ISH):

If the split horizon rule is used, the source loops marked with “ISH” never occur because they are no composite back-routes $m_{A,B}^{i,d,i}$.

Proof: Two route combinations $m_A^{i,d}$ contain a route $m_A^{i,d}$ with the corresponding path

$$P_A^{i,d} = (i, \dots, A, i, A, \dots, d) \text{ or } P_A^{i,d} = (i, \dots, B, i, B, \dots, d)$$

That means that the interface A (B) could serve as in and out interface in the path $P_A^{i,d}$. These composite back-routes are avoided **by the source router i** which applies the split horizon rule (“A router does not send outgoing routing updates back to the router from which it has learned this route”). In a path

$$P^{i,d} = (r_1 = i, s_1, r_2, s_2, \dots, r_l, s_l = d),$$

the condition $s_x \neq s_{x+1}$ holds for any router r_x ($1 \leq x \leq l-1$).

Theorem 3 (External Split Horizon, ESH):

If the split horizon rule is used, the source loops marked with “ESH” never occur because they are no composite back-routes $m_{A,B}^{i,d,i}$.

Proof: Three composite back-routes $m_{A,B}^{i,d,i}$ contain a route $m_A^{i,d}$ with the corresponding path $P_A^{i,d} = (r_1 = i, s_1 = A, r_2 \in NR(A), \dots, r_{t-1} \in NR(A), s_{t-1} = A, r_t = i, \dots, d)$ with $NR(A) = \{\text{neighbor routers of interface } A\}$. These composite back-routes are avoided by the neighbor routers applying the split horizon rule. They do not send those routes, which they have learned from router i , back to router i , because router i compared with the neighbor routers holds the shortest metric to destination d , and therefore router i always is the next router for destination d in the routing tables of the neighbor routers.

Avoiding X Combinations

Fig. 6 gives an example for a composite back-route with an X combination. In this case there is a composite back-route between interfaces A and B .

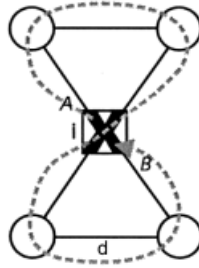


Fig. 6. Example of a composite back-route with an X combination.

From the viewpoint of router i the part of the AS including interface A and the part of the AS including interface B are separate. To get from the one part of the AS to the other, the source router i has to be traversed. It would not make any sense to use a route via interface A as an alternative route to destination d reachable via interface B . X combinations are constructed by concatenating two different cycles, in this case the upper and lower cycle. The X combination would leave the router via interface A , take a return route back to the source router i , enter the router via a interface different to A , leave the source router once again via another interface other than A or B and at last return via interface B . X combinations can be avoided by making sure that the metric of the route combination is less than the sum of the two minimal return routes via the first (A) and the second interface (B).

Theorem 4:

Let $m_{A,B}^{i,d,i}$ be a composite back-route. If $\min m_A^i + \min m_B^i > m_{A,B}^{i,d,i}$ then the composite back-route $m_{A,B}^{i,d,i}$ does not contain an X combination.

Proof: A composite back-route with an X combination has a path

$$P_{A,B}^{i,d,i} = (i, A, \dots, s_1 \neq A, i, s_2 \neq B, \dots, d, \dots, B, i)$$

The first part of this path, from the first occurrence of i to the second occurrence of i , is a return route via interface A . The second part of the path, from the second occurrence of i to the third occurrence of i is a return route via interface B . Therefore, the shortest X combination is the sum of the minimal return routes via interface A and B ($\min m_A^i + \min m_B^i$).

If the composite back-route $m_{A,B}^{i,d,i}$ is shorter than the shortest possible X combination, then it cannot contain an X combination.

Avoiding Y Combinations

Fig. 7 shows an example of a composite back-route with a Y combination. The problem with Y combinations is as follows. Router i knows the route to destination d via interface B , because it is directly connected to i . If router $r1$ learns from $r2$ that it can reach d via $r2$, $r1$ will send an update to router i containing $\langle m_A^{r1,d} = 3 \rangle$.

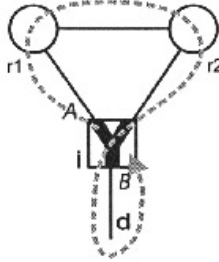


Fig. 7.Example of a composite back-route with a Y combination.

Router i then has to decide whether d can be reached via interface A too. The route via interface B is correct, but the route via interface A contains a source loop, which is constructed as follows: the route leaves router i via interface A , returns to i it via a different interface and takes the first route via interface B directly to d .

The minimal return route via interface A advertising the alternative route with the higher metric plays the crucial role here. Y combinations can be avoided by applying the following condition:

Theorem 5:

Let $m_{A,B}^{i,d,i}$ be a composite back-route combining the two route advertisements $m_A^{i,d}$ and $m_B^{i,d}$; $m_B^{i,d} \geq m_A^{i,d}$. If $\min m_A^i > m_A^{i,d} - m_B^{i,d}$ then the composite back-route does not contain a Y combination.

Proof: The longer route $m_{A,B}^{i,d,i}$ within $m_{A,B}^{i,d,i}$ belongs to the path $P_A^{i,d} = (i, A, \dots, i, B, \dots, d)$. The first part of this path from the first to the second occurrence of i is a return route via interface A and the second part is the

corresponding path of $m_B^{i,d}$. Therefore $m_A^{i,d}$ is the sum of any return route via interface A and $m_B^{i,d}$.

If $m_A^{i,d}$ is smaller than the sum of the minimal return route via interface A and $m_B^{i,d}$:

$$m_A^{i,d} < \min m_A^i + m_B^{i,d}$$

then $m_A^{i,d}$ does not contain a return route via A and $m_{A,B}^{i,d,i}$ is not a source loop.

3.3 The RIP-MTI Algorithm

In an acyclic AS it is not possible for any router to reach an arbitrary destination by two different routes over two different interfaces. If a router has a choice between different interfaces to a destination d , the AS must be cyclic, with the result that after the failure of a route to destination d it can possibly be replaced by an alternative route. But after such a failure the router must not accept any route to destination d as an alternative route: if the route is a member of a source loop this may lead to “counting to infinity”.

With theorem 6 it is possible to decide whether two routes concatenated to build a composite back-route form a source loop or a cycle.

Theorem 6 (Corollary):

Let $m_{A,B}^{i,d,i}$ be a composite back-route and $m_A^{i,d} \geq m_B^{i,d}$. All routers use the “split horizon” rule. If

a) $\min m_A^i + \min m_B^i > m_{A,B}^{i,d,i}$ (to avoid X combinations) and

b) $\min m_A^i > m_A^{i,d} - m_B^{i,d}$ (to avoid Y combinations),

then $m_{A,B}^{i,d,i}$ is a cycle $\text{cyc}_{A,B}^{i,d,i}$ and therefore neither a source loop nor a routing loop and no “counting to infinity” can arise.

3.4 Calculating Minimal Return Routes

Up to this point, the minimal return $\min m_A^i$ route via an interface A was assumed to be given. By looking at the construction of the X and Y combination composite back-routes in detail it can be seen that in most of the cases it is possible to use the composite back-route metric $m_{A,B}^{i,d,i}$ as upper bound of $\min m_A^i$ and $\min m_B^i$. Only in the case of a Y combination can the composite back-route metric not be used as an upper bound of $\min m_B^i$ where $m_A^{i,d} \geq m_B^{i,d}$.

3.5 Overcoming Minimum-Hop Routing

The problem with using a non-minimum-hop-metric is, that the formula for calculating composite back-routes ($m_{A,B}^{i,d,i} = m_A^{i,d} + m_B^{i,d} - 1 = L$) can be used for minimum hop metrics only. One way to overcome this problem is to extend the routing update message and the routing tables, with the hop count metric as additional information besides the actual metric. The RIP-MTI algorithm is executed with this hop count metric only calculating minimal cycles and minimal return routes using the previously described method. In this case, extending the routing protocol cannot be avoided.

4 Simulating RIP-MTI

4.1 Simulation Environment

To simulate RIP-MTI a Java Applet² was developed that allows the graphical building of models of networks, and enables switching between the original RIP and the new RIP-MTI algorithm [6]. During simulation connections between interfaces and networks can be cut, which makes it possible to provoke a “counting to infinity” situation. The temporal order in which the routers send their periodic routing updates is crucial for the emergence of counting to infinity. Hence the simulation program allows timing to be influenced.

For analysis two different models were chosen (see fig. 8) representing the models in fig. 7 and fig. 6.

4.2 Simulation Results

Corresponding to the two simulated models shown in fig. 8 every possible temporal order was simulated with small and large periods between the routers leaving out those orders symmetrically equivalent to already simulated orders.

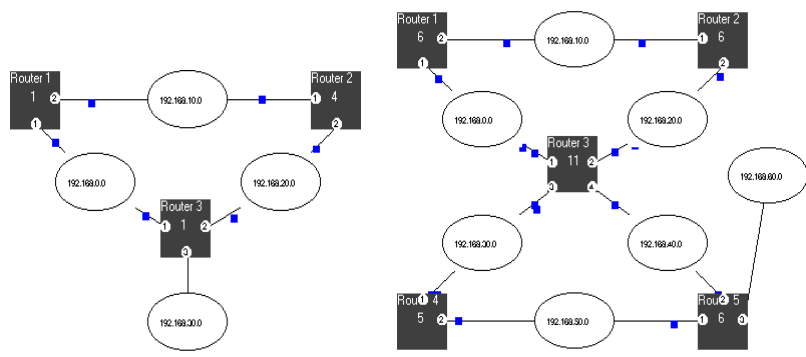


Fig. 8. Simulating a Y and X combination

² <http://www.uni-koblenz.de/~steigner/ripmti/>

There were 86 counting to infinity situations in total when using the original RIP algorithm (36 in model 1 and 50 in model 2). All of them were avoided when switching to the RIP-MTI algorithm resulting in much faster convergence.

Fig. 9 shows how many routing updates were needed in the network in a counting to infinity situation to reach convergence using RIP-MTI compared to RIP with “infinity” set to 16, 31 and 61. The numbered labels on connectors depict two temporary orders in model 1 and four temporary orders in model 2.

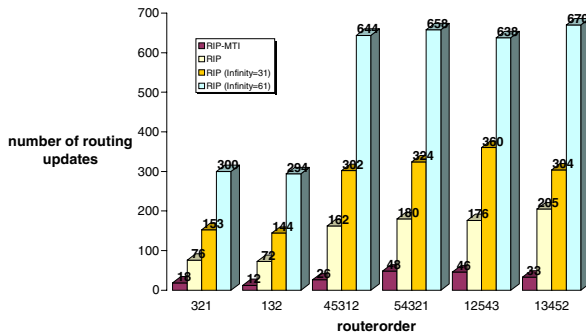


Fig. 9. RIP-MTI comparative convergence behavior

RIP-MTI accelerates convergence by 73 to 83% compared with the original RIP algorithm. When increasing infinity first to 31 and then to 61 improvements of 85-91% and 92-95% respectively were seen. This is because an increased value for “infinity” does not influence convergence in RIP-MTI. These factors can be transferred to the reduction of network traffic and to the duration until convergence, because they depend on the number of routing updates needed.

5 Conclusions

This paper showed that source loops are the cause of routing loops, which are responsible for counting to infinity. Therefore two conditions to avoid source loops were given and proven. These conditions were integrated into the extended RIP algorithm RIP-MTI, storing information about cycles between every pair of local interfaces of a router, which is a new and unique approach. Finally it has been shown by simulation that RIP-MTI does exactly what has been proved in theory: routing loops and counting to infinity behavior are eliminated resulting in much faster convergence with minimal demand for additional memory and processing power.

References

- [1] R. Braden and J. Postel, *Requirements for internet gateways*, IETF RFC 1009, USC/Information Sciences Institute, June 1987
- [2] C. Cheng, R. Riley, S. P. R. Kumar and J. J. Garcia-Luna-Aceves, *A loop-free extended bellman-ford routing protocol without bouncing effect*, in ACM SIGCOMM Symposium Commun. Arch. and Protocols, pp. 224-236, Austin, Texas
- [3] J. J. Garcia-Luna-Aceves, *A unified approach to loop-free routing using distance vectors or link states*, in ACM SIGCOMM Symposium Commun. Arch. and Protocols, pp. 212-223, Austin, Texas, Sept. 1989
- [4] Garcia-Luna-Aceves (1993), *Loop-free routing using diffusing computations*, IEEE/ACM Transactions on Networking 1, No. 1, pp. 130-141
- [5] J. M. Jaffe and F. M. Moss, *A responsive routing algorithm for computer networks*, IEEE Transactions on Communications COM-30, No. 7, pp. 1785-1762, July 1982
- [6] O. Kandel, *RIP-MTI – Simulation des erweiterten RIP-Algorithmus*, Diploma thesis, University of Koblenz-Landau, Institute for Computer Science, Rheinau 1, 56075 Koblenz, Germany, July 2000 (German)
- [7] B. Rajagopalan and M. Faiman, *A new responsive distributed shortest path routing algorithm*, in ACM SIGCOMM Symposium Commun. Arch. and Protocols, pp. 237-246, Austin, Texas, Sept. 1989
- [8] G. C. Sackett and C. Metz, *ATM and multiprotocol networking*, McGraw-Hill, 1996
- [9] A. J. Schmid, *Solution for the counting to infinity problem of distance vector routing*, Research report 7-98, University of Koblenz-Landau, Institute for Computer Science, Rheinau 1, 56075 Koblenz, Germany, June 1998, URL <http://www.uni-koblenz.de/fb4/publikationen/gelbereihe/RR-7-98.ps.gz>
- [10] A. J. Schmid, *RIP-MTI: Minimum-effort loop-free distance vector routing algorithm*, Diploma thesis, University of Koblenz-Landau, Institute for Computer Science, Rheinau 1, 56075 Koblenz, Germany, March 1999
- [11] A. J. Schmid and C. Steigner, *A solution for the counting to infinity problem of distance vector routing*, Proc. of the Second IASTED International Conference on Parallel and Distributed Computing and Networks (Brisbane, Queensland, Australia), December 1998, pp. 471-477
- [12] K. G. Shin and M.-S. Chen, *Performance analysis of distributed routing strategies free of ping-pong-type looping*, IEEE Transactions on Computers C-36, No. 2, pp. 870-881, Feb. 1987
- [13] A. S. Tanenbaum, *Computer Networks*, Prentice Hall, third edition, 1996

Proposal of an Inter-AS Policy Routing and a Flow Pricing Scheme to Improve ASes' Profits

Nagao Ogino and Masatoshi Suzuki

KDD R&D Laboratories Inc.,
2-1-15 Ohara, Kamifukuoka-shi,
Saitama 356-8502 Japan
{ogino, suzuki}@kddlabs.co.jp

Abstract. At present, the global Internet consists of many ASes. Each AS pays a pre-determined connection fee to another AS for connecting its network with another AS's network. The connection fee type charging may be rational in case of transferring the best-effort type traffic. However, usage charging is necessary to transferring the resource guaranteed type traffic such as the Intserv traffic and the Diffserv traffic. In this case, each AS pays a per-flow fee to another AS every time it routes a flow into another AS. The per-flow fees paid by each AS becomes a part of the cost for that AS. Thus, each AS needs to select a route with the lowest price for each inter-AS flow to improve its profit. In this paper, we call such an inter-AS routing scheme a price-based inter-AS routing scheme. When each AS has a request to route an inter-AS flow, it can select an inter-AS route with the lowest price to improve its profit by this routing scheme. Firstly, we propose a method to realize the price-based inter-AS routing scheme. Next, we propose cost-dependent pricing scheme suitable for the price-based inter-AS routing scheme. The cost-dependent pricing scheme can reduce frequency of exchanging price information between ASes. However, in the cost-dependent pricing scheme, profit in each AS depends on the distribution of path costs in that AS. Thus, we propose a routing policy for ASes with narrow ranges of path costs to improve their profits efficiently and verify its effect using a simple routing model.

1 Introduction

At present, the global Internet consists of many ASes. Here, AS (Autonomous System) means network provider with original network control policy to maximize his/her profit. Each AS pays a pre-determined connection fee to another AS for connecting its network with another AS's network. The connection fee type charging may be rational in case of transferring the best-effort type traffic. However, it causes a serious problem in case of transferring the resource guaranteed type traffic such as the Intserv traffic [1] and the Diffserv traffic [2]. If the connection fee is constant, an AS may transfer vast traffic into another AS and network resources in the latter AS may be occupied by only the traffic transferred from the former AS. In this way, the

constant connection fee results in lack of network resources and unfairness between ASes [3].

Usage charging is necessary to solve the problem described above. Each AS pays a per-flow fee to another AS every time it routes a flow into another AS. Each AS can manage every inter-AS flow well because the number of inter-AS flows becomes relatively small by aggregating many flows between a source AS and a destination AS into a jumbo flow [4]. Accounting of packets conveyed on a flow may be necessary to realize the usage charging strictly. However, for example, contracted bandwidth and duration time of a flow can be substituted for number of packets in case of the expedited service and the assured service mentioned in the Diffserv specification [2]. Of course, price rate of flow should be varied according to the class of service that the flow belongs to.

The per-flow fees paid by each AS becomes a part of the cost for that AS. Thus, each AS needs to select a route with the lowest price for each inter-AS flow to improve its profit. In this paper, we call such an inter-AS routing scheme a price-based inter-AS routing scheme. The price-based inter-AS routing scheme can be realized by exchanging route price information between ASes. For example, price-vector information needs to be exchanged in addition to the path-vector information in the case of inter-AS routing protocol such as the BGP-4 (Border Gateway Protocol -4) [5].

In this paper, we propose a method to realize the price-based inter-AS routing scheme. When each AS has a request to route an inter-AS flow, it can select an inter-AS route with the lowest price to improve its profit by this routing scheme. Next, we propose cost-dependent pricing scheme suitable for the price-based inter-AS routing scheme. The cost-dependent pricing scheme can reduce frequency of exchanging price information between ASes. However, in the cost-dependent pricing scheme, profit in each AS depends on the distribution of path costs in that AS. We propose a routing policy for ASes with narrow ranges of path costs to improve their profits efficiently and verify its effect using a simple routing model.

2 Price-Based Inter-AS Routing Scheme

The price-based inter-AS routing scheme requests exchange of route price information between ASes. For example, price-vector information needs to be exchanged in addition to the path-vector information in the case of inter-AS routing protocol such as the BGP-4 [5]. Fig. 1 explains the price-based inter-AS routing scheme using the distance vector type routing protocol. Here, price of a path is defined as usage fee per unit of contracted bandwidth and per unit of duration time charged for flows which traverse that path. Of course, the price of a path depends on the class of service that the flow belongs to.

The price of a path in each AS (P_x) corresponds to sum of the path cost and the profit for that AS. It can be regarded as a distance of that AS (d_x) in case of the distance vector type routing protocol. Thus, distance of each route (R_d) becomes total

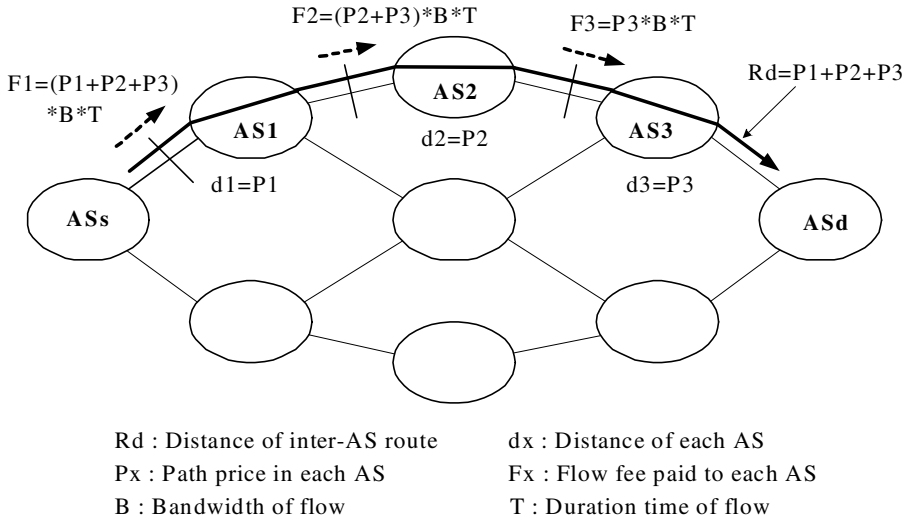


Fig. 1. Price-based inter-AS routing scheme

sum of path prices in all the ASes composing that route. Each AS floods total distance, i.e., sum of its own distance and the route distance received from the neighboring downstream AS, to the neighboring upstream AS. In other words, each AS calculates sum of the path price in that AS and the route price received from the neighboring downstream AS. If this newly calculated value is less than total route price calculated previously, each AS floods this value to the neighboring upstream AS as new total route price. Each AS holds a price-vector table including information of each destination AS and the least total route price to that AS. Each AS can select an inter-AS route with the lowest price by this routing scheme when it has a request to route an inter-AS flow. By this way, each AS can reduce the cost caused by the per-flow fee that the AS has to pay the downstream AS, and can improve its own profit.

Each AS needs to pay the per-flow fee corresponding to the total price charged by all the downstream ASes that convey the flow in sequence. The per-flow fee paid by each AS is distributed to all of its downstream ASes. Fig. 1 also shows a method for distributing the per-flow fee to the individual downstream ASes. In this method, each AS claims the per-flow fee (Fx) corresponding to the total price charged by itself and all of its downstream ASes to the neighboring upstream AS. Next, each AS pays the per-flow fee claimed by the neighboring downstream AS to that AS. In other words, each AS obtains a part of the per-flow fee received from the neighboring upstream AS, and pays the remained part of the per-flow fee to the neighboring downstream AS. The per-flow fee which each AS should claim to the neighboring upstream AS can be calculated easily using the price-vector table that each AS holds. Of course, the per-flow fee is accumulated e.g. during a month, and those claim and payment of the per-flow fee are performed once per month between two ASes.

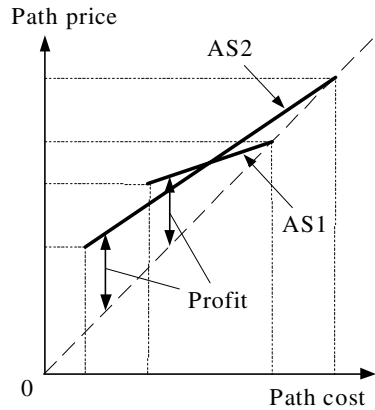


Fig. 2. Cost-dependent pricing in two ASes

3 Pricing Scheme in Each AS

State-dependent pricing can be considered as a pricing scheme for each path in the AS [6][7]. In the state-dependent pricing scheme, price generally increases as path utilization increases. By this way, congestion can be avoided and users are given incentives to select a route with the highest QoS. However, the price-vector information needs to be exchanged between ASes frequently because the price of a path varies according to the change of its utilization.

Here, we propose cost-dependent pricing suitable for the price-based inter-AS routing scheme. In the cost-dependent pricing scheme, price is slightly reduced when path cost is small. As is shown in Fig. 2, this pricing scheme gives users incentives to select a route with low cost and thus ASes can obtain larger profit. Here, we consider profit obtained from a flow per unit of bandwidth and during a unit of time. Therefore, the profit can be defined as difference between the path price and the path cost. If the route cost reflects static QoS like the number of hops along the route, the cost-dependent pricing may give users incentives to select a route with higher static QoS.

The path cost is generally constant. Therefore, the price of a path is also constant in the cost-dependent pricing. This means that the cost-dependent pricing is more suitable for the price-based inter-AS routing scheme because it can reduce frequency of price information exchanges between ASes. Hereafter, we investigate the price-based inter-AS routing scheme in the case where each AS adopts the cost-dependent pricing scheme.

4 Routing Policy for ASes

In the cost-dependent pricing scheme, profit that each AS can obtain may be restricted by distribution of path costs in that AS. For example, let us consider an inter-AS route traversing two ASes with different ranges of path costs, as is shown in Fig. 2. The average path costs of AS1 and AS2 are identical, and profit levels in both ASes are also identical. Though AS1 and AS2 have the same price level, AS2 has a wider range of path costs and thus a wider range of path prices compared with AS1. It is assumed that the path costs in both ASes follow the uniform distribution within their ranges.

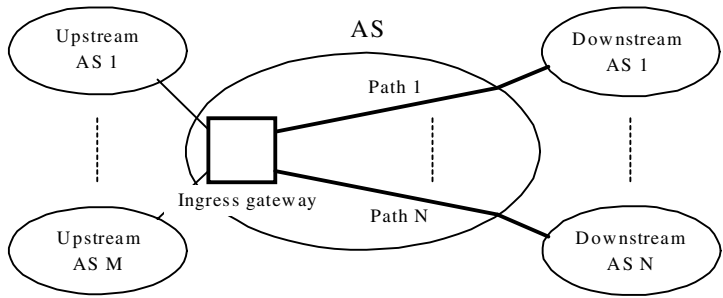
In the case of Fig. 2, AS2 has larger influence on the total price of the inter-AS route. In other words, the inter-AS route is usually selected because of low path price in AS2. This means that AS1 cannot obtain large profit compared with AS2. On the other hand, AS1 cannot expand the price range unlimitedly because at least the price range must be smaller than the cost range.

We propose a routing policy for ASes with narrow ranges of path costs to improve their profits efficiently. In this routing policy, each AS prohibits flows from which that AS cannot obtain profit more than “shadow price” [8]. Here, the shadow price means profit that the AS can expect to obtain using unutilized resource resulting from rejection of the flow. In other words, each AS permits only flows by which that AS can increase its long-term profit.

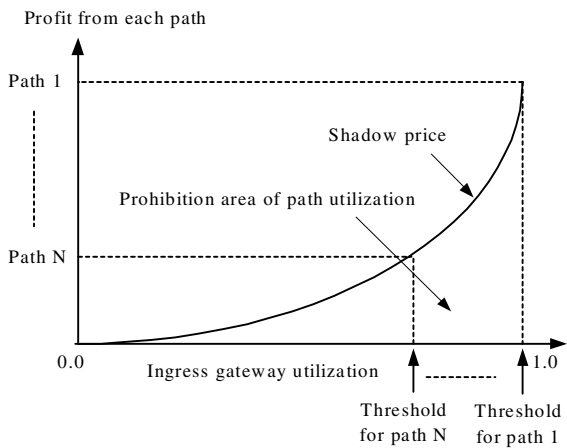
Profit from a flow is generally constant in each AS because path cost and thus path price for that flow is invariable in each AS. On the other hand, the shadow price in each AS depends on only flow arrival rate and utilization rate in the path that the flow traverses. This means that flows traversing a path are prohibited when utilization of that path is larger than a threshold given in advance. In other words, each AS floods an prohibition signal to other ASes only when the path utilization exceeds the threshold, and floods a permission signal only when the path utilization falls below the threshold. This threshold corresponds to the path utilization rate where the profit obtained from the path and the shadow price of the path are identical with each other.

Because each AS floods signals only when the path utilization crosses the threshold, this routing policy does not increase frequency of information exchanges between ASes compared to the state-dependent pricing scheme. Prohibiting flows from traversing a path corresponds to regarding the price of that path as infinite. For this reason, the proposed routing policy can be realized within the framework of price-vector type inter-AS routing protocol.

Fig. 3 shows an example of the proposed routing policy. In Fig. 3 (a), the ingress gateway is the bottleneck resource for each path, and the utilization of the ingress gateway determines the shadow prices of all the paths. If flow arrival rates of all the paths are identical, the shadow price of each path also becomes identical. Fig.3 (b) shows the relationship between the shadow price and the profit obtained from each path. When the gateway utilization exceeds threshold for a path, flows are prohibited from traversing that path because that path gives only profit less than the shadow price.



(a) Example of internal structure of an AS



(b) Example of relationship between profit from each path and shadow price

Fig. 3. An example of routing policy

5 Performance Evaluation of Routing Policy

5.1 Performance Evaluation Model

Here, it is assumed that the flow arrival is random and the flow duration time follows negative exponential distribution. Moreover, each inter-AS route that a flow traverses is assumed to be unchanged during the existence of that flow. This means that packets of a flow are conveyed on a fixed connection such as LSP (Label Switched Path) [9]. From those assumptions, the value of shadow price and economical efficiency of the proposed routing policy can be analyzed using the policy iteration method derived from the Markov decision theory [10][11]. In the policy iteration method, the policy is updated repeatedly until it converges on the optimum policy.

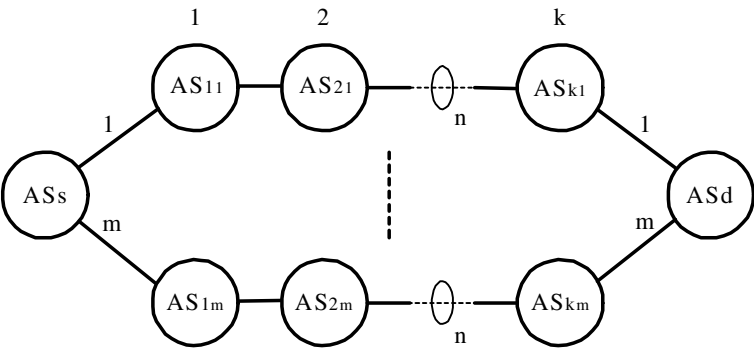


Fig. 4. Routing model for evaluating routing policy

However, we have to update policies for more than one AS in the inter-AS routing model. There may exist no Nash equilibrium point in the inter-AS routing model. Moreover, the policy iteration method does not assure convergence on a Nash equilibrium point even if a Nash equilibrium point exists. In this paper, the routing policies for ASes are updated alternately and the update of policies continues until the total profit of all the ASes becomes maximum. In other words, alternate policy update is finished if the total profit of all the ASes decreases by updating the policy.

In this paper, only several AS-disjoint inter-AS routes between a pair of source AS and destination AS are considered as is shown in Fig. 4. Those routes are assumed to traverse the same types of ASes each other and have an identical bandwidth. Bandwidth utilizations in those routes are assumed to be independent of each other. Thus, the amount of calculation can be reduced. In Fig. 4, there exist m disjoint routes with n units of bandwidth between the source AS and the destination AS. Each route consists of k ASes. It is assumed that every flow requests a unit of bandwidth. Here, we denote the applied traffic intensity per unit of bandwidth by A .

We consider five types of ASes with the same average path cost but with different ranges of path costs. Those ASes are assumed to have an identical profit level and an identical price level as a result of economic competition between ASes. Nevertheless, they have different ranges of path prices because they have different ranges of path costs. Table 1 shows the path cost range and the path price range in each AS type.

The path cost in each AS is given randomly and uniformly within each path cost range when a new inter-AS flow arrives. In other words, each AS has virtually infinite number of paths. All the paths in each AS are assumed to share a bottleneck resource and utilization of this bottleneck resource corresponds to the path utilization in all the paths. The path utilization in the ASes composing an inter-AS route is assumed to be identical and this path utilization corresponds to the bandwidth utilization in that route.

Table 1. Path cost and path price in each AS

AS type	Path cost		Path price	
	Minimum	Maximum	Minimum	Maximum
1	4.0	5.0	5.0	5.0
2	3.0	6.0	4.0	6.0
3	2.0	7.0	3.0	7.0
4	1.0	8.0	2.0	8.0
5	0.0	9.0	1.0	9.0

Table 2. Composition of routes utilized for each experiment

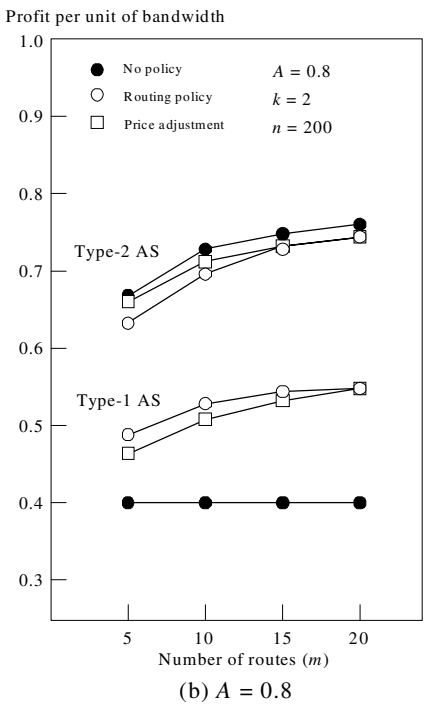
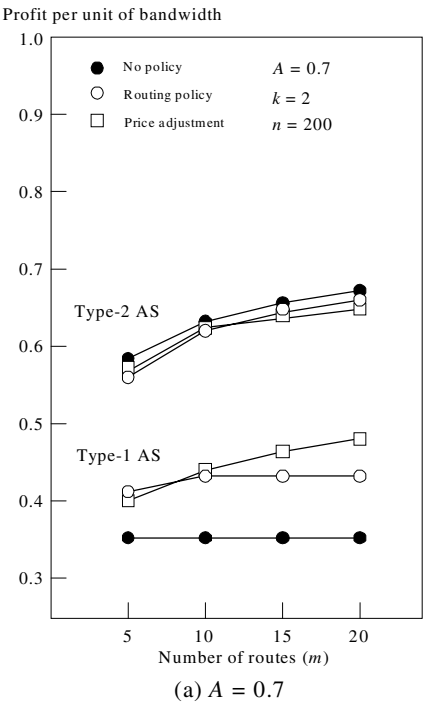
Experiment	AS types
I	$1 (+ 1 (+ 1)) + 2$
II	$1 + 1 + 2 + 2 , 1 + 1 + 2 + 3 ,$ $1 + 1 + 2 + 4 , 1 + 1 + 2 + 5$

Table 2 shows composition of inter-AS routes utilized in the following experiments I and II. The experiment I evaluates the effect of routing policy when the number of inter-AS routes and the number of ASes composing an inter-AS route vary. The experiment II evaluates the effect of routing policy when various types of ASes compose each inter-AS route.

5.2 Results of Experiment I

Fig. 5 shows relationship between the profit per unit of bandwidth in each AS and the number of routes m . In Fig. 5, each route consists of one type-1 AS and one type-2 AS ($k = 2$), and have 200 units of bandwidth ($n = 200$). The profit in the type-1 AS is smaller than that in the type-2 AS because the path price range in the type-1 AS is narrower than that in the type-2 AS. However, the type-1 AS can improve its profit by adopting the routing policy proposed in this paper. Fig. 5 shows the profit in each AS when the sum of profit in two types of ASes becomes maximum by the alternate update of the routing policies for two ASes.

If the rates of price change to cost change are identical in two ASes, a route with the least cost can be selected by the price-based inter-AS routing scheme. For the purpose of comparing the effect of price adjustment with that of the routing policy, Fig. 5 also shows the profit in each AS when a route with the least cost is selected by adjusting the path prices in the type-1 AS.



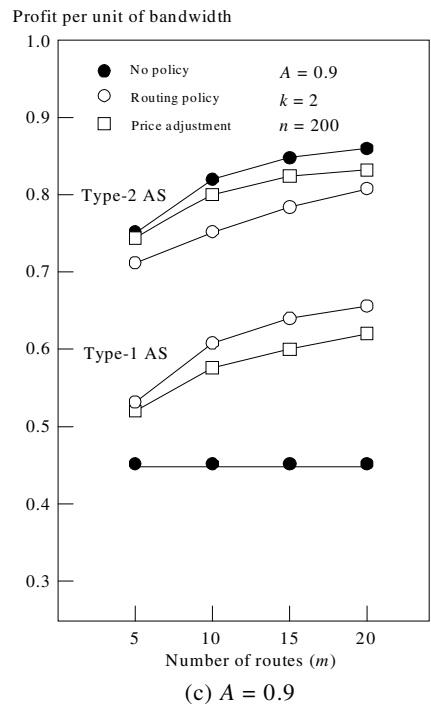


Fig. 5. Profit versus number of routes

Except for the type-1 AS in the “No policy” and the “Routing policy”, the profit in each AS becomes larger as the number of routes increases. This result is caused by the cost-dependent pricing scheme. In other words, an inter-AS route with less price can be selected as the number of routes increases, and this inter-AS route with less price can give more profit to each AS in the cost-dependent pricing scheme.

By adopting the routing policy, the type-1 AS can improve its profit drastically while the profit in the type-2 AS is slightly reduced. This improvement in the profit becomes larger as the value of A becomes larger, compared with the improvement caused by the price adjustment. This is because the shadow price becomes larger by the increase of the applied traffic intensity. At this time, only flows giving larger profit are permitted by the routing policy in the type-1 AS. The probability that all the routes prohibit a flow is less than four percent at every case in Fig. 5.

Fig. 6 shows effect of the routing policy when the number of type-1 ASes involved in each route varies. In Fig. 6, the number of routes is fixed at 10 and each route has 200 units of bandwidth. Fig. 6 shows the profit in each type of AS when the total sum of profit in all the ASes becomes maximum by adopting the routing policy. As the number of type-1 ASes increases, effect of the routing policy is reduced because many type-1 ASes must share the effect. In other words, the improvement of profit in the type-1 AS decreases and the reduction of profit in the type-2 AS increases as the number of type-1 ASes increases.

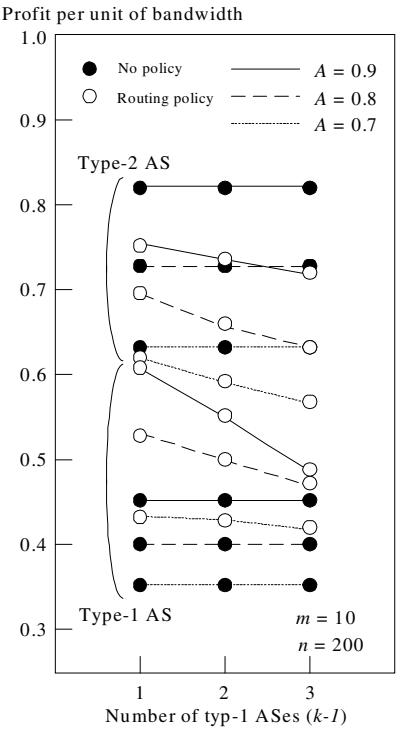


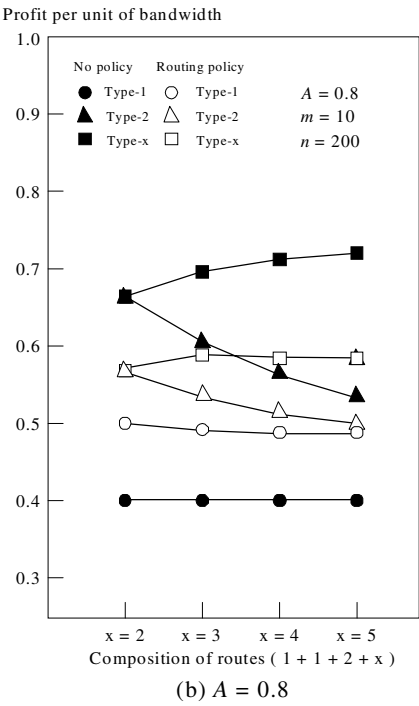
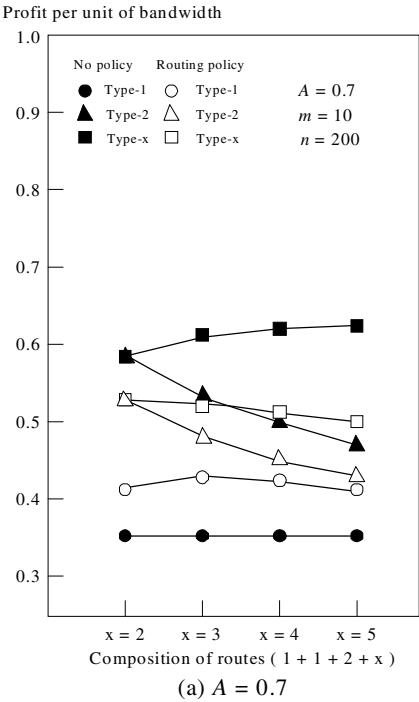
Fig. 6. Profit versus number of type-1 ASes

5.3 Results of Experiment II

Fig. 7 shows effect of the routing policy when the composition of routes varies. Each route is composed of two type-1 ASes, one type-2 AS, and one type- x ($x = 2 \sim 5$) AS. The number of routes is 10 and each route has 200 units of bandwidth. Fig. 7 shows the profit in each type of AS when the sum of profit excluding the type- x AS becomes maximum by adopting the routing policy.

As is shown in Fig. 7, effect of the routing policy is reduced as the path price range in the type- x AS expands. This is because the path price in the type- x AS becomes too dominant on the whole route price when the path price range in the type- x AS is wide. At this time, the routing policy cannot compensate for the profit in the type-1 AS sufficiently. Since the type-2 AS has the middle price range, the profit in the type-2 AS also decreases by adopting the routing policy. However, this decrease is smaller than that in the type- x AS.

AS a conclusion, the ASes with narrow ranges of path costs can improve their profits efficiently by adopting the routing policy, so long as no AS with extreme dominance on the whole route price exists.



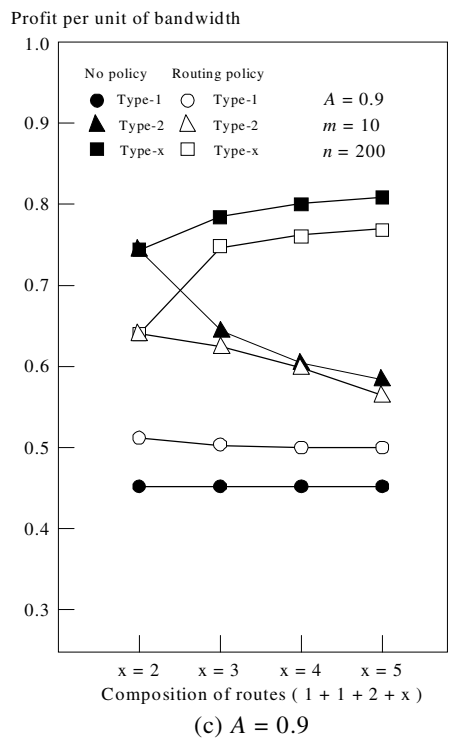


Fig. 7. Profit versus composition of routes

6 Conclusions

We proposed a method to realize a price-based inter-AS routing scheme, where price for a path in each AS is regarded as a distance of that AS. When each AS has a request to route an inter-AS flow, it can select an inter-AS route with the lowest price to improve its profit by this routing scheme. Next, we proposed cost-dependent pricing scheme suitable for the price-based inter-AS routing scheme. The cost-dependent pricing scheme can reduce frequency of exchanging price information between ASes. In the cost-dependent pricing scheme, profit in each AS depends on the distribution of path costs in that AS. We proposed a routing policy for ASes with narrow ranges of path costs to improve their profits efficiently and verified its effect using a simple routing model. The ASes with narrow ranges of path costs can improve their profits efficiently by adopting this routing policy, so long as no AS with extreme dominance on the whole route price exists.

The price-based inter-AS routing scheme needs to be evaluated using more practical routing model. Game theoretical analysis on the price-based inter-AS routing scheme is also necessary. Those items are left for further studies.

Acknowledgments

We would like to express our gratitude to Dr. Akiba, President & CEO, Dr. Asami, Executive Vice President, and Dr. Matsushima, Vice President of KDD R & D Laboratories Inc., for their encouragement throughout the study.

References

1. Braden, R., Clark, D., and Shenker, S.: Integrated Services in the Internet Architecture: an Overview. IETF RFC 1633 (June 1994)
2. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W.: An Architecture for Differentiated Services. IETF RFC 2475 (Dec. 1998)
3. MacKie-Mason, J. K. and Varian, H. R.: Pricing Congestible Network Resources. IEEE JSAC, Vol. 13, No. 7 (Sept. 1995) 1141-1149
4. Fang, W. and Peterson, L.: Inter-AS Traffic Patterns and Their Implications. Proc. of IEEE Globecom '99 (Dec. 1999) 1859-1868
5. Rekhter, K. and Li, T.: A Border Gateway Protocol 4 (BGP -4). IETF RFC 1771 (March 1995)
6. MacKie-Mason, J. K., Murphy, L., and Murphy, J.: Responsive Pricing in the Internet. In Internet Economics. MIT Press (1997) 279-303
7. Kuwabara, K., Ishida, T., Nishibe, Y., and Suda, T.: An Equilibrium Market-based Approach for Distributed Resource Allocation and Its Application to Communication Network Control. In Clearwater, S. H. (ed): Market-based Control: A Paradigm for Distributed Resource Allocation. World Scientific (1996)
8. Kelly, F. P.: Routing in Circuit-switched Networks: Optimization, Shadow Prices and Decentralization. Adv. Appl. Prob., Vol. 20 (1988) 112-144
9. Rosen, E., Viswanathan, A., and Callon, R.: Multiprotocol Label Switching Architecture. IETF RFC 3031 (Jan. 2001)
10. Dzong, Z. and Mason, L. G.: Call Admission and Routing in Multi-service Loss Networks. IEEE Trans. on Communications, Vol. 42, No. 2/3/4 (Feb./March/April 1994) 2011-2022
11. Ross, S. M.: Introduction to Stochastic Dynamic Programming. Academic Press, Inc. (1983)

Stigmergic Techniques for Solving Multi-constraint Routing for Packet Networks

T. Michalareas¹ and L. Sacks²

Telecommunication System Research Group
Electrical & Electronics Department
University College London (UCL), London, UK,
`t.michalareas,l.sacks@ee.ucl.ac.uk`

Abstract. In this paper we describe how stigmergic techniques can be used in packet networks that offer soft QoS services. The problem we are interested is the on-line version of computing routes to be established over a packet network, and the number of constraints imposed by the service is more than one. We investigate the scheme of the algorithm, the issues around the characteristics of the constraints and we give some simulation evidence of the working algorithm.

1 Introduction

In current IP networks the offered service is best-effort as there are no guaranteed bounds on the end-to-end delivery of packets (in terms of delay, jitter, economic cost of any other metric). One of the challenges of the evolving IP technology is to offer such guarantees with the minimum cost for the network. Apart from the forwarding function that needs to be enhanced inside the router engines, the routing algorithms that compute the paths the traffic packets follow end-to-end has to take into account the type of constraints those packets require. Such ToS(Type of Service) or DSCP(Differentiated Services Code Point) aware algorithms would give benefits in terms of resource utilization at the core of the network.

The version of the problem we are interested is a variance of QoS-aware routing. The first main assumption is that the algorithm does not use any explicit knowledge of reservation of resources across network nodes. Such algorithms could operate in both a network that performs plain best-effort hop-by-hop forwarding, and any other that uses Differentiated Services forwarding mechanisms. The second assumption concern the form of constraints and the expectations of the applications we consider across the QoS spectrum. At one edge of the spectrum there are application that can operate well with best-effort service (e.g. e-mail and FTP). At the other edge there are application that they can benefit out of resource reservations that would guarantee them their stringent margins of expected network performance (e.g. telephone calls, video-conferencing). In

between there are applications that have less marginal expectations from end-to-end performance. Such applications would benefit out of any possible optimization of their considered metrics but they can also adapt. We term the area in the middle of the QoS spectrum, soft QoS.

In our case we consider a system that supports a single soft-QoS service (although the proposed algorithm can easily be extended to support more than a single service). We will consider two constraints. The first considered constraint is the average delay for packets. The second constraint is the available capacity of the network along an end-to-end connection. The routes are determined in a hop-by-hop manner. The problem of QoS-aware routing has been proven to be NP-complete in [6]. A number of algorithms that use heuristics have been proposed to solve relaxed version of the problem [6], [7],[8]. In this paper we examine a swarm intelligence approach into solving the problem.

2 Swarm Intelligence in Networks

In [1][2] [3] there have been presented distributed algorithms for computing routes in cases of traditional telephone networks and best effort (symmetric or asymmetric) IP networks based on the biological paradigm of ants foraging behaviour. The technique used by ant colonies to locate and transfer food supplies into their nest, or even construct complex structures has been termed *stigmergy* [4] and it has been a source of inspiration for computer scientist investigating discrete optimization problems like TSP [5].

The proper definition of stigmergy by Grasse is the following : "Stimulation of workers by the performance they have achieved" So basically stigmergy is a positive feedback mechanism using chemical substances like pheromones to attract agents, that themselves depose these chemicals. That loop reinforces solutions selected by the majority of the biological agents and even with random initial conditions (that is with agents selecting among viable paths with equal probability) optimal solutions (shortest paths) can be found for transporting food back to the nest. This capacity of estimating shortest paths has been at the core of the algorithm suggested by various computer scientists.

The agents instantiate themselves in the virtual space of routes, exploring various possibilities, until the system finds one that cannot be reinforced any more. A system that uses solely positive feedback could lock to poor solutions rather than the global optima. There is a need for a negative feedback mechanism that would stop this unwilling "freeze" of the system. In the case of real pheromones this role is performed by the physical evaporation. In the case of the routing algorithms different algorithms have used different techniques. All of the algorithms share the use of virtual "chemicals", "pheromones" to act as intermediates for stigmergy. These pheromones are usually treated as probabilities.

The "swarm" of agents uses a number of feedback mechanisms to influence the final paths selected for forwarding packets.

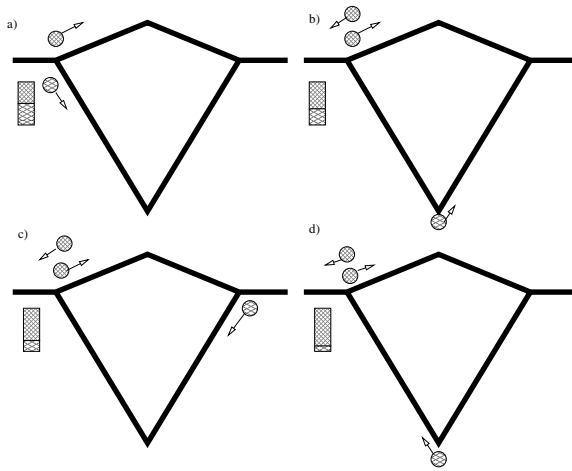


Fig. 1. Implicit negative feedback mechanism in operation among ant-like agents in a decision point. The volume of the pheromone along a path is affected by the number of agents that traverse it, that depends on the pheromone along the path and the metric value of a path. The shortest path is traversed on average by more agents, double the number of agents that follow the alternative path in this simple example where the length of the shortest path is half the length of the alternative.

Positive Feedback This mechanism is used to reinforce the use of one of the alternate paths. Upon the arrival of a B_{ant} to a network node the p_m metric value measured by the corresponding F_{ant} is used to produce a raw value r' . The r' value is used to increase the probability of the interface the F_{ant} used to get to that node.

Negative Feedback This mechanism is responsible for decreasing the probability of an alternate path to be selected. There are two negative feedback mechanisms during the operation of a single swarm. The first is due to the use of the pheromone vectors as probability distribution. That means the pheromones' sum as in Eq.1 should always be equal to 1. The increase of a pheromone by the positive feedback, necessitates the decrease of all the others. The second negative feedback mechanism is implicit. The delay of F_{ant} to a node accordingly to the metric value of the sub-path it has experienced gives the advantage to the agent that have followed the best available sub-path up to that node. The corresponding B_{ant} has the chance to increase the probability of its followed path to be selected, before the ones of the F_{ant} that have followed an alternative route. The net effect is the reduction of the volume of ants following alternative routes (see Fig.1).

The proposed algorithms tackle the problem of on-line routing for networks that support a single type of service that has a single constraint. In the telephone network the service is the establishment of a call and the constraint is that the selected routes should minimize the probability of a call being rejected for the

system. In the packet network case the service is the delivery of packets from a source to a destination in the “best-effort” sense. The constraint is that the routes selected should minimize the average delay the packets experience traversing the network.

3 Multi Constraint Algorithm

3.1 General Scheme

The scheme we describe in this section is not based on the net effect of a single virtual “Swarm” (termed also Ant Colony or the set of agents) but to the combined effect of multiple Swarms. For that reason we term our algorithm as Multi-Swarm. The main idea is to associate with each unique constraint-metric pair m a unique Swarm $Swarm_m$ that has its own defined behaviour and with each service a type of pheromone. The Swarms that are “stimulated” by constraints that are included in a service definition change the unique pheromone vectors that correspond to that service (see Fig.2).

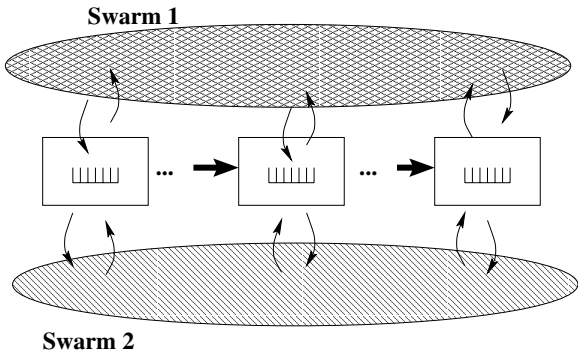


Fig. 2. Principle of operation for the Multi-Swarm Routing Algorithm, Both Swarms affect the same pheromone, which is an expression of the probability of a path selected for the service.

Each swarm uses the idea of “virtual pheromones”. In traditional routing tables for every destination a single entry is kept that indicates which of the possible nodes’ interfaces should be used to forward the packet. The routing tables that are used in our algorithm hold for every destination node a vector that has the size of the number of node’s interfaces. The values hold there, are restricted in the $[0,1]$ interval and there are termed pheromones, due to the way there are updated by algorithms agents. The packets of the routing protocol are forwarded probabilistically. These virtual pheromones can be treated for forwarding user packets in two ways.

In stochastic forwarding the vector is treated as a probability distribution over the interfaces for sending packets towards a particular destination. That

means that the interface that is selected for each packet to be forwarded towards a specific destination follows that distribution. The deterministic way uses a number of rules to determine a number of interfaces to be used. In the single path version of the problem a single interface should be eliminated (e.g. the one with maximum pheromone tables). In the multi-path version a number of interfaces are used by utilizing hash functions to determine which subsets of packets moving towards a destination will use one of these interfaces.

There are two sets of virtual agents (termed also “swarms of ants”) that are responsible for the update of the pheromone tables. Each swarm takes decisions based on the single constraint that is associated with and updates the pheromones’ routing table. There are two forms of implicit communication (via the pheromone values). The agents of each swarm affect fellow agents that will traverse the node in future time. They also affect agents of other swarms that will traverse the link.

For individual swarms we follow the ideas presented in [2] in order to deal with the asymmetric nature of traffic and both the statistic and non-stationary variations of the metrics used. The mechanism is extended to deal with non-additive metrics.

3.2 Additive Metric

The mechanism described in the following paragraphs takes as an example the delay metric, but it can be applied for any additive metric. Every node with id k holds a list $Trip_k(\mu_i, \sigma_i)$ of estimates of the arithmetic mean μ_i and the associated variances σ_i for trip times from that node to all others in the network. This data structure is the view a node has for the delay reaching any other node. The $Trip_k$ values are used when the nodes update the pheromone tables along with measurements from the agents.

To update its data structures in an asymmetrical network every node has two types of agents. The forward ant agent $F_{s \rightarrow d}$ is spawned in regular intervals from all possible nodes s towards destinations nodes d , which are selected randomly. Every forward ant has a memory stack. The forward ants are forwarded towards their destination using the pheromone tables and they use the same queues as all other packets. Whenever a node receives a forward ant, it checks whether it has arrived in this node before and if it had (cycle detection) it removes all the entries of the cycle from ant’s stack. Then it pushes the arrival time into the agent’s memory stack.

When a forward ant reaches its destination it spawns a backward ant agent $B_{s \rightarrow d}$ at which it transfers its memory and terminates. The backward ant follows the reverse path the stack indicates and use its values to update the $Trip_k(\mu_i, \sigma_i)$ data structure and the pheromone table of every intermediate node.

When it arrives at the source of the initial forward ant it terminates. The update rules for the pheromone table have to conserve the sum of the every vector goodness values to 1 (see Equation-1).

$$\sum_{n \in N_k} = 1, i \in [1, N], N_k = neighbors(k) \quad (1)$$

We follow the update rules from [2], estimating a new parameter r' whenever a backward ant reaches a node, coming from a node f . This r' value is used to update the goodness values of the vector that is associated with the source node of the backward ant. The entry of the vector that corresponds to the interface that the associated forward ant used in its journey is increased (rule $r+$ in Equation-2) and the other are decreased (rule $r-$ in Equation-3) where P_{df} and P_{dn} are the last probability values assigned to neighbors of node k for destination d .

$$r_+ = (1 - r') * (1 - P_{df}) \quad (2)$$

$$r_- = (1 - r') * P_{dn}, n \in N_k, n \neq f \quad (3)$$

The estimation of r' involves a number of steps. Initially a raw estimation of r' is computed using Equation-4. In Equation-4, T is the observed trip time of the ant, μ is its mean value as stored in list $Trip_k(\mu_i, \sigma_i)$. This rule saturates out of range values to 1. The second step depends on the comparison of σ/μ with a arbitrary small threshold ε . If σ/μ is less that the threshold value, the algorithm considers that the observations on the mean are stable. In this case the previously computed value of r' is decreased or increased by the quantity $S(\sigma, \mu, a)$ (as in [2]).

$$r' = \begin{cases} \frac{T}{c\mu}, c \geq 1 & \text{if } \frac{T}{c\mu} < 1 \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

At the other case the quantity is increased or decreased by the quantity $U(\sigma, \mu, a')$ (as in [2]) where $a' \leq a$. Finally the value obtained at the second step is filtered through a power law and bounded in the interval of $[0,1]$. The reasoning for the second step is that the algorithm should discriminate the case where the traffic fluctuations follow the same statistics and the case that this statistics change. At the first case poor r' values are increased and good r' values are decreased. At the second case, where the algorithm cannot consider r' a reliable measure of goodness, it tries to find a solution by amplifying the good r' values and suppressing the poor ones. For more observations on the behaviour of Ant-like algorithms and the semantics of the parameter values look at [2][9].

3.3 Concave Metric

The other constraint that we are interested in, is the available bandwidth. This constraint is not additive. The metric of a path is the minimum metric of the links it is comprised of and not the addition of them. The non-additive nature of the metric necessitates another mechanism for updating the pheromones. Another

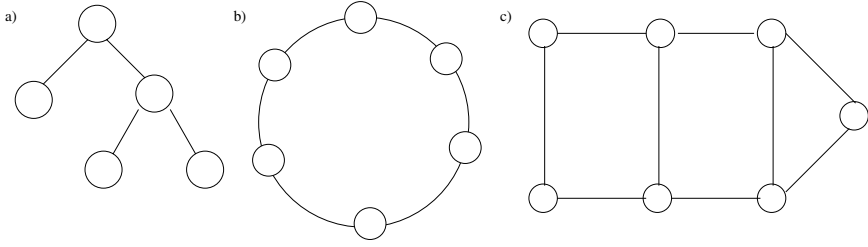


Fig. 3. *Topologies Used for Simulation Scenarios*

difference of this metric from delay, is that it cannot be measured by the agent itself. In the case of the delay metric the agent could register its departure and arrival time and estimate the effective delay it experienced through the link. This is not possible for available bandwidth. A resource monitor should be present at either side of the link to report the link available bandwidth. Another side-effect is that the arrival rate of the agents belonging to the swarm corresponding to available bandwidth cannot be used the same way as in the mechanism for the delay metric.

1. The fact that the metric is concave reduces the necessary stack space to a single cell that would hold the minimum value of available bandwidth, S experienced by the agent over its path. So to cope with the concave nature of the metric we reduce the number of values stored over the Forward Ant of the Swarm adventure in the network.
2. We introduce a resource monitor that the agent communicates with, at its arrival to a node. The resource monitor holds both average and deviation values for the utilization of the links of all incoming interfaces.
3. Since using the agents arrival rate the same way as for the delay swarm is not indicative of how many available bandwidth they have encountered, we introduce an artificial delay in its node to restore the validity of that information. The Forward agents are delayed in a similar way to the ABC algorithm in [1]. The proposed delay for an agent traversing a link with spare capacity S is given in Equation-5. The value of r' is computed by the formula in Equation-6, where C is the capacity of the link.

$$D = c * e^{-d.S} \quad (5)$$

$$r' = \frac{a}{C - S} + b \quad (6)$$

In order to test the operation of our scheme we have simulated a single-constraint version of the algorithm over different topologies with uniform load.

4 The Simulation Scenario

In our simulation we use three simple network topologies to estimate how different is the behaviour of the Ant-like algorithm compared to a Link State algorithm at steady state. In our case we used as an example of Link State algorithm, LS that uses Dijkstra algorithm internally to compute routes. For comparison reasons we also use a static routing algorithm called Session routing.

The topologies considered, is a Tree, a Cycle and a General graph. The Tree is a topology that offers no alternative routes for any node in the network. The Cycle topology is the other extreme that offers the same number of alternative route for every node as interfaces. The example of General Graph is an intermediate situation where the nodes have connectivity between 2 and 3. The topologies are as in Figure-3.

The LS and the Session algorithm have no parameters associated with them. The Ant-like algorithm has a number of parameters associated with the discovering properties like the parameters c, a, a', ε, h (the exponential of the power law). In our simulations we kept the same parameters values as in [2]. In our scenario the traffic is generated by a number of FTP sessions established between every possible source destination source. We vary the number of active sessions between source and destination pairs according to a uniform distribution with minimum parameter a 0 and a varying maximum parameter b . Because we are interested in the steady state performance of the algorithm all connections start simultaneously after 100sec from the beginning of the simulation. Every simulation lasts for 100sec.

5 Statistical Methodology and Measurements

In order to validate our results statistically we repeated five (5) times every experiment with a different seed. We selected their values based on the internal seed values that the ns-2 Simulator is using if the user selects a heuristic way to set the values. The distances between the random seeds are close to one million. We are interested at the total packets that have been successfully forwarded under a routing scheme. To measure that for every simulation we sample the number of successfully acknowledged packets from the TCP layer every 0.05 seconds during the 100sec long period the sessions are active.

For steady state behaviour we are interested in averaging out the behaviour of TCP sessions due to the initial condition and the Slow-start phase. To succeed that we also sample the current window size of the TCP layer every 0.05 seconds. In our simulations we use TCP Tahoe. In this version TCP increases the window size exponentially during its Slow-start phase and linearly at the congestion avoidance phase. We are interesting in identifying the end of the first Slow-start phase that is due to the beginning of the simulation. We use the differential of the window size for that purpose. We chop our statistical sample until the differential is stable.

To compare the behaviour of the algorithms we estimate the total throughput of each of our experiments for each of them. We estimate the average of the

total throughput over the different samples. We also use the advanced Anderson-Darling(A2) test rather than the commonly used Kolmogorov-Smirnov test of normality(see [10] for reasoning) to check whether the total throughput measurements fir the normal curve. We also estimate the 95% confidence interval for every scenario as we increase the average load.

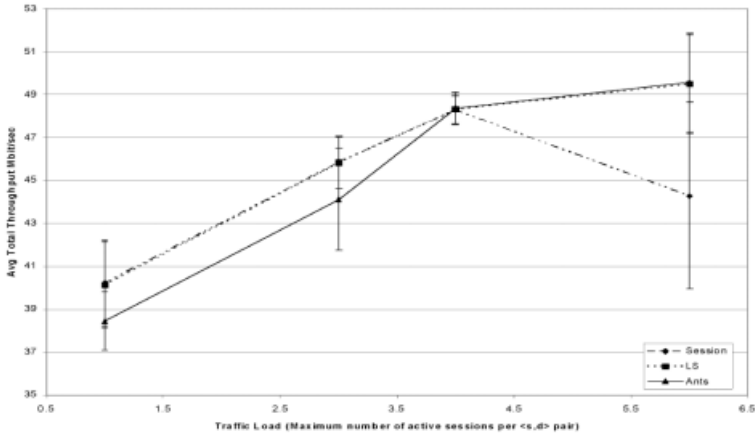


Fig. 4. Comparison in terms of Total Throughput for the Tree Topology

6 Results

All our Anderson-Darling tests of normality showed a reasonably good fit of our data to the normal curve distribution. That means that we can use the $z_{.025}$ value for estimating the 95% confidence interval for our data. The measurements are summarized in Figure-4, Figure-5, Figure-6. At the y-axis we show the achieved throughput in Mbits/sec for a specific scenario. At the x-axis we show the maximum number of active sessions for each source-destination pair in the network. The y-axis error bars show the 95% confidence intervals of average total throughput over the 5 samples we have for every scenario). The results for the other topologies follow the same trends.

We can see that the routing schemes have similar achieved throughput measurements. The behaviour of the averaged total throughput of the network is always between the 95% confidence interval zone.

7 Summary

We have presented the Multi-Swarm algorithm , an Ant Colony inspired algorithm to compute routes based on more than a single constraint. The algorithm

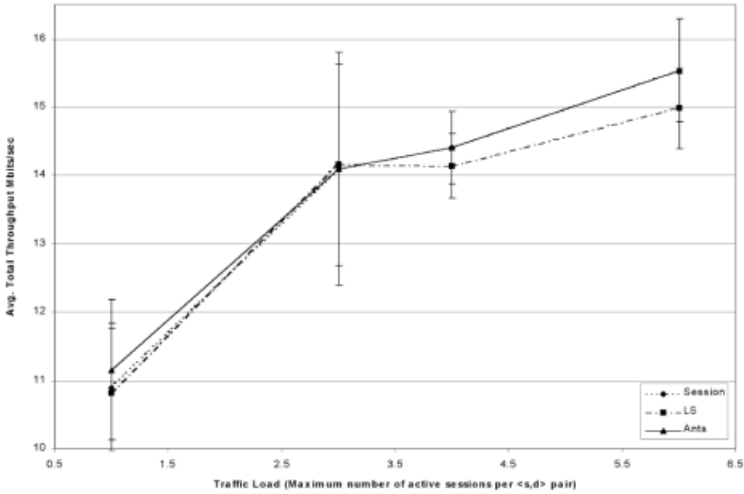


Fig. 5. Comparison in terms of Total Throughput for the Cycle Topology

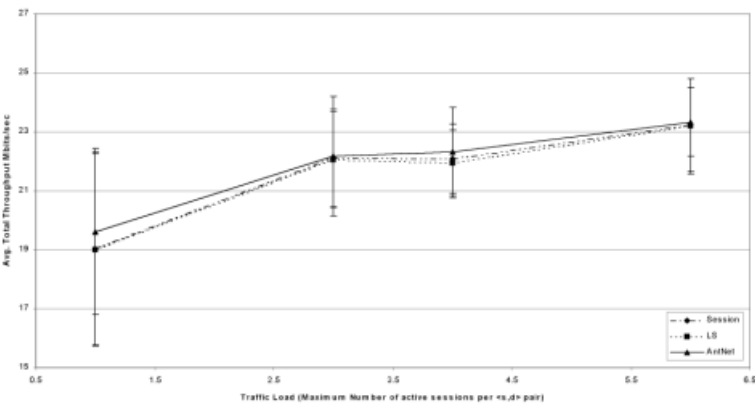


Fig. 6. Comparison in terms of Total Throughput for the Graph Topology

also can be used as a protocol that collects network statistics using for that purpose the capabilities of the agents. Although no capabilities of active networking or agent platform are necessary for such a scheme, it can physically be integrated to such architectures of network management. We have tested its behaviour in steady state under uniform TCP traffic, in terms of the achieved throughput. The results show similar performance to that of Link-State algorithm, that cannot scale to more that two independent constraints(as in [6], [7], [8]) and suffer when link-state update frequencies is large.

References

1. Schoonderwoerd, R., O. Holland, J. Bruten and L. Rothkrantz. “*Ant-based Load Balancing in Telecommunications Networks.*”, *Adapt. Behav.* 5 (1996),pp169-207.
2. Di Caro, G., and M. Dorigo. “*AntNet: Distributed Stigmergetic Control for Communication Networks.*”, *Journal of Artificial Intelligence Research*, in press.
3. Heusse, M., S. Guerin, D. Snyers and P. Kuntz. “*Adaptive Agent-Driven Routing and Load Balancing in Communication Networks.*”, Technical Report RR-98001-IASC, ENST Bretagne, Brest, France, 1998.
4. Grasse, P. “*La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensins et cubitermes sp. La theorie de la stigmergie : essai d’interpretation du comportement des termites constructeurs.*”, *Insectes Sociaux*, 6:41-81,1959.
5. Dorigo, M., and L. M. Gambardella. “*Ant Colony System : A Cooperative Learning Approach to the Travelling Salesman Problem.*”, *IEEE Trans. Evol. Comp.* 1 (1997): pp 53-66.
6. Wang, Z., and Jon Crowcroft, “*Quality of Service Routing for Supporting Multimedia Applications.*”, *IEEE JSAC*, 14(7): pp. 1288-1234, Sep. 1996.
7. G. Apostolopoulos, et. al., “*QoS Routing Mechanism and OSPF Extensions.*”, IETF RFC 2676, Aug. 1999, <http://www.ietf.org/rfc/rfc2676.txt>.
8. Dean H. Lorenz, Ariel Orda, “*QoS Routing in Networks with Uncertain Parameters.*”, *IEEE/ACM Transactions on Networking*, Vol. 6, No. 6, Dec. 1998,pp.768-778
9. Michalareas, T., “*Stigmergy for Routing and Network Management.*”, Transfer Thesis for Ph.D. at EE Department, UCL, London, August 2000.
10. Vern Paxson, “*Empirically-Derived Analytic Models of Wide-Area TCP Connections : Extended Report*”, June 1993, UC, Lawrence Berkeley Laboratory

Dynamic Capacity Resizing of Virtual Backbone Networks

Seung H. Rhee¹, Jaewoo Yoon², Heonjun Choi², and Insoo Choi²

¹ Kwangwoon University, Seoul, Korea,
rhee@ieee.org

² ETRI/National Security Research Institute, Daejeon, Korea,
{jyoon,chj,ischoi}@nsri.etri.re.kr

Abstract. A virtual private network (VPN) functionality should include a performance guarantee provided to the customers. To provide guaranteed services, the network provider allocates appropriate capacities to multiple virtual backbone networks such that the underlying network can be shared among them. As VPN users are demanding reliable and dynamic allocation of capacities, recently the capacity resizing approach has been considered as a cost efficient way of providing virtual network services. We propose a new scheme for dynamic allocation of virtual link capacities. The allocated capacities are adjusted dynamically according to the users' requests such that their capacities are increased in a fair manner and the total reservation does not overwhelm the underlying network. Depending on the network's status and allocation policy, a virtual link may increase or decrease its capacity, for example, for a monetary incentive. VPN users send request packets whenever they want to resize their capacities, and the network handles them in an efficient and fair way. The simulation and analytic results shows that our scheme is simple and robust such that the link capacities are allocated in an efficient and fair manner.

1 Introduction

A virtual private network (VPN) service is likely to be used by customers as a replacement for networks constructed using private lines[1], and thus its functionality should include the performance guarantee provided to those customers. As many of VPN customers' applications will not allow the delay and packet loss in the public Internet, their virtual backbone links require some performance assurances. Such virtual links can be realized by packet scheduling algorithms, label switched paths, or other layer 2 tunneling mechanisms. Most of the previous VPN mechanisms, however, paid much less attention to resource management issues. In order to achieve the statistical multiplexing in the provider's network, and thus increase the utility of the underlying network and give the VPN users some economical benefits, the capacity of virtual links should be dynamically adjusted. We expect that it is difficult for the VPN customers to specify their QoS requirements exactly, and they might want to change the capacities of the virtual backbones occasionally, or in some case frequently. A couple of approaches

on the dynamic resource management in VPN can be found in literature: In [2], a new service interface is introduced to provide the dependable and dynamic connectivity between end points, and [3] has proposed a similar approach which provides dynamic capacity allocations to different classes.

In this paper, we develop a new capacity resizing mechanism for dynamically adjusting the virtual links. We are not concerned with the problems of how to provide the bandwidth guarantees or how to match users' requirements and service levels. Rather we are interested in how to resize the virtual link capacities in a fair and efficient manner. In our scheme, VPN customers can request to resize their virtual links by sending control packets to the network manager, and according to the network status and the allocation policy, they are allowed to resize their virtual backbones. Thus after establishing a virtual link, the capacity is not permanent and can be dynamically updated according to the traffic demand of the VPN customer. The network provider handles the requests such that the total reservation does not overwhelm the underlying network and the residual capacity, if available, is fairly distributed among competing customers.

This paper is organized as follows. The model is formulated in Section 2 and its property is examined and simulated in Section 3. In Section 4 we discuss the extension to the case of multi-hop virtual links, and Section 5 concludes the paper.

2 The Model

We first describe the framework for the single link case. A set $\mathcal{K} = \{1, \dots, K\}$ of virtual links(VL) is given and share the link of capacity C . Initially capacity A_i is allocated to virtual link i such that $\sum_{i \in \mathcal{K}} A_i \leq C$. A virtual link(VL) can request to increase or decrease its reservation, and the capacity resizing server(CRS), the link management entity, adjusts the reservation to r_i according to the admission rule described below. Presumably the VPN user will have some monetary incentive to voluntarily decrease its reservation for the VL. We note that, while it may represent the customer-pipe of a VPN, each endpoint of a VL may be characterized by the *hose*[2], which specifies the capacity required for aggregate incoming/outgoing traffic at the endpoint.

VL(Virtual link) behavior

A VPN user who owns a VL generates a special packet that contains a desired capacity s_i that the user wants to reserve and the current reservation r_i . After returning back the packet from the capacity resizing server(CRS), it can increase (or decrease) its reservation to s_i which is assigned by the CRS. After setting $r_i = s_i$, if $r_i \leq A_i$, that is, current reservation is less than or equal to the initially-allocated capacity, then it can be maintained until the user wants to change the capacity again. However, if $r_i > A_i$, the VL generates the control packet periodically so that its capacity is dynamically changed according to the network status and CRS's decisions.

Capacity Resizing Server(CRS) behavior

Define two disjoint sets of VPN users $\mathcal{K}_1$ and $\mathcal{K}_2$ as the users with $r_i \leq A_i$ and $r_i > A_i$, respectively. CRS is responsible for keeping track of the residual capacity of the link and for distributing it to those VLs who want to change their reservations. It maintains three kinds of information:

1. $\tilde{C}$, the available capacity of the link; initially, it is given by $\tilde{C} = C - \sum_i A_i$, and as VLs update their capacities,

$$\tilde{C} = C - \sum_i A_i + \sum_{i \in \mathcal{K}_1} (A_i - r_i)$$

2. A_i for all $i \in \mathcal{K}$
3. The summation R of excess capacities of all VLs in $\mathcal{K}_2$; i.e., if we let $f_i \equiv r_i - A_i$ for $i \in \mathcal{K}_2$, then

$$R = \sum_{i \in \mathcal{K}_2} f_i = \sum_{i \in \mathcal{K}_2} (r_i - A_i)$$

That is, f_i is the amount of excess bandwidth above the initial allocation A_i , and R is the summation of f_i for all VLs in $\mathcal{K}_2$. $\square$

When CRS receives a control packet, it compares s_i in the packet with A_i and handles the request as follows.

Case 1 If $s_i \leq A_i$, then the request is always accepted regardless of the current reservations, and the information is updated by

$$R = R - f_i \cdot I(i \in \mathcal{K}_2) \quad \text{and} \quad \tilde{C} = \tilde{C} + ((A_i \wedge r_i) - s_i),$$

where $I(\cdot)$ is the indicator function and $a \wedge b = \min(a, b)$. Thus, if the previous reservation of VL i is greater than A_i , then both of R and $\tilde{C}$ are updated; Otherwise, only $\tilde{C}$ is changed.

Case 2 If $s_i > A_i$, the decision is made according to the value of r_i as follows. If $r_i \geq A_i$, then a new capacity is assigned by

$$s_i = s_i \wedge \{A_i + \alpha(\tilde{C} - R + f_i)\}, \quad (1)$$

where $0 < \alpha < 1$. The rationale behind the equation (1) and the role of α in distributing the residual capacity will be discussed in Section ?? . Else if $r_i < A_i$, then first increase r_i up to A_i and let $\tilde{C} = \max\{0, \tilde{C} - (A_i - r_i)\}$. Now we have the same case as the above and repeat the allocation by (1). Thus the VL i received at least A_i and s_i is determined by the network status. $\tilde{C}$ and R are updated according to the change of r_i .

After the admission, or even during the process of the second case, the total reservation may exceed the link capacity such that $\sum_{i \in \mathcal{K}} r_i > C$. However, since VPN users in $\mathcal{K}_2$ are assumed to generate the control packets continuously and

the excess capacity allocated to those users is distributed from the residual capacity $\tilde{C}$, that situation may exist only for a very short period of time. $\square$

Note that the control scheme described above is similar to the rate-based ABR traffic control mechanism in ATM networks [4]. The distinct aspect of our scheme is that the increase or decrease of VL capacity is motivated by users and the management entity accept or reject their requests according to the admission policy. In the ABR control, by contrast, the flow rates are controlled by the network, not by the users. Moreover, our scheme allows the users to reduce their reservations below their initial allocated capacities.

3 Applying the Allocation Policy

3.1 Analysis of the Allocation Policy

In order to analyze the stability and property of the allocation policy described in the previous section, we only consider the VLs with $s_i > A_i$. In this case, regardless of the current reservation r_i , the VL capacity is resized by (1) and decisions are made such that $s_i \geq A_i$. Note that, if $s_i \leq A_i$, then the allocation is deterministic and no algorithm is required. Thus, if $\tilde{C}$ is given, the allocation problem is reduced to the case where the VLs in $\mathcal{K}_2$ compete for the excess capacity. In a real situation, the VLs and the residual capacity can be dynamically changed, and thus the allocation algorithm should be stable and fair to distribute the resource among the VLs. As the users in $\mathcal{K}_2$ are assumed to send control packets, f_i is updated periodically as long as $\tilde{C}$ changes.

Now consider a system of optimization problems below:

$$\begin{aligned} & \text{maximize} && U_i(f_i, i \in \mathcal{K}_2), \quad \forall i \in \mathcal{K}_2 \\ & \text{subject to} && 0 \leq f_i \leq s_i - A_i, \end{aligned} \quad (2)$$

where U_i is a real function that is strictly concave with respect to f_i and has its optimal at

$$\arg \max_{f_i} U_i(f_i, i \in \mathcal{K}_2) = \alpha(\tilde{C} - \sum_{j \neq i} f_j). \quad (3)$$

Each user in the system of (2) are trying to optimize their utilities, and the optimal policy of a user given a constant $\tilde{C}$ and other users' policies is always determined by (3). The modeling and analysis of a system of optimizations applied in the area of communication networks was studied in [5], and it has been known that the resource allocation problem such as the above has a unique equilibrium such that all users are satisfied at the point.

Now consider our resizing scheme. If the CRS assigns f_i according to (3) and the constraint of (2), then it corresponds to the optimal policy of virtual link i . Thus, if the CRS determines f_i for all users in $\mathcal{K}_2$ in the same way, then it corresponds to the situation formulated in (2). According to the unique equilibrium property of the model, it follows that 2 has a unique solution which

is fair to all users in $\mathcal{K}_2$. Thus, the allocation is fair and stable as long as $\tilde{C}$ remains constant.

Moreover, the optimal solution can be achieved dynamically using the *Gauss-Seidel* type iterations[6]; that is, the network handles users' requests one at a time either synchronously or asynchronously. It has been known that if the some updates are made at the same time, then they might not converge to an equilibrium[7]. However, in our scheme, even if the control packets from VLs arrive at the same time, the capacity updates are computed one at a time and they are guaranteed to converge. Also note that VLs need no knowledge on the link capacity and other VLs; As long as the control packet are delivered without an error, the transmission delay of the packets between the users and network management entity does not exert any bad influence on the convergence.

We now analyze the capacity resizing achieved at the equilibrium. Assuming VLs in $\mathcal{K}_2$ are competing for a residual capacity $\tilde{C}$, let f_i^* and F^* be the allocated capacity for VL i and the total allocation from $\tilde{C}$ at the equilibrium, respectively. The actual reserved capacity of VL i is then $r_i = A_i + f_i^*$. The following theorem tells how much capacity each VL can get out of $\tilde{C}$.

Theorem 1. *Let $s'_i = s_i - A_i$, where $s_i, s_i > A_i$, is the desired capacity of VL $i \in \mathcal{K}_2$. If a residual capacity $\tilde{C}$ is distributed with the parameter α , $0 < \alpha < 1$, then the capacity f_i^* allocated to $i \in \mathcal{K}_2$ at the equilibrium is determined such as*

$$f^{*i} = \begin{cases} s'_i & , \text{ if } s'_i < FS \\ FS & , \text{ otherwise,} \end{cases}$$

where the unique point FS , a fair share, is given by

$$FS = \frac{\alpha}{1-\alpha}(\tilde{C} - F^*). \quad (4)$$

Proof. By applying the Kuhn-Tucker conditions [8], we have the following conditions for all i at the equilibrium of (2):

$$\lambda_i^* \geq 0, \mu_i^* \geq 0, \lambda_i^*(f_i^* - s'_i) = 0, \mu_i^* f_i^* = 0 \quad (5)$$

$$-\alpha(\tilde{C} - \sum_{j \neq i} f_j^*) + f_i^* + \lambda_i^* - \mu_i^* = 0, \quad (6)$$

where λ_i^* and μ_i^* are Lagrange multipliers. We have the following from (5):

$$\lambda_i^* < \mu_i^* \Rightarrow \mu_i^* > 0 \Rightarrow f_i^* = 0$$

$$\lambda_i^* > \mu_i^* \Rightarrow \lambda_i^* > 0 \Rightarrow f_i^* = s'_i$$

Then (6) can be written as

$$f_i^* - \frac{\alpha}{1-\alpha}(\tilde{C} - F^*) + \frac{1}{1-\alpha}(\lambda_i^* - \mu_i^*) = 0 \quad (7)$$

Now let $FS = \frac{\alpha}{1-\alpha}(\tilde{C} - F^*)$, then from (7) and above implications,

$$s'_i < FS \Rightarrow f_i^* < FS \Rightarrow \lambda_i^* > \mu_i^* \Rightarrow f_i^* = s'_i$$

The remaining case $f_i^* = FS$ can be proved by removing the users of the first case and computing the unconstrained equilibrium as in [9]: Define $\mathcal{A}$ and $\mathcal{K}'_2$ to be $\mathcal{A} = \{i \in \mathcal{K}_2 | s'_i > FS\}$ and $\mathcal{K}'_2 = \mathcal{K}_2 - \mathcal{A}$, respectively. Further, let

$$\tilde{C}' = \tilde{C} - \sum_{i \in \mathcal{A}} s'_i.$$

Then for $i \in \mathcal{K}'_2$,

$$f_i^* = \frac{\alpha}{1 + \alpha(K'_2 - 1)} \tilde{C}' = \alpha \tilde{C}' - f_i^* \alpha (K'_2 - 1), \quad (8)$$

where $K'_2 = |\mathcal{K}'_2|$. Summing up for all $i \in \mathcal{K}'_2$ yields $F' = K'_2 \alpha \tilde{C}' - F' \alpha (K'_2 - 1)$, where $F' = \sum_{i \in \mathcal{K}'_2} f_i^*$. We have $\alpha K'_2 = \frac{F'}{\tilde{C}' - F'}$, from this equation and substituting it into (8) completes the proof of the other case as follows:

$$f_i^* = \frac{\alpha}{1 - \alpha} (\tilde{C}' - F') = \frac{\alpha}{1 - \alpha} (\tilde{C} - F^*). \quad \square$$

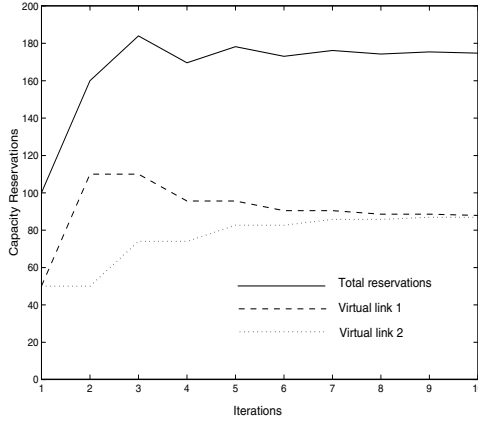


Fig. 1. Dynamic updates of VL capacities: two VLs with constant available capacity

3.2 Capacity Resizing in a Single Link

In this section, we simulate the capacity resizing policy. Fig. 1 shows the capacity resizing of two VLs. Initially the two VLs are provisioned with $C_1 = 50$ and $C_2 = 50$. The situation assumes that the two VLs are competing for the residual capacity $\tilde{C} = 100$. They continues to send control packets and, after a number of interactions with the CRS, their reservations converge to the point where the

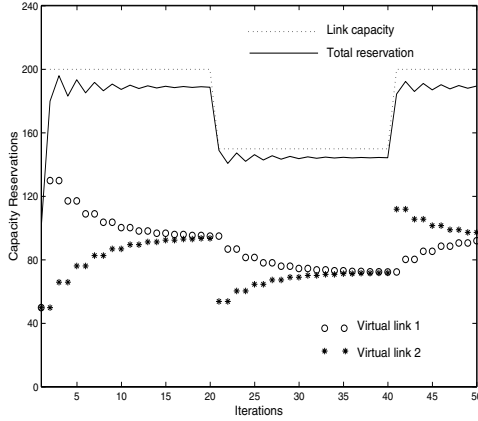


Fig. 2. *Dynamic updates of capacity reservations: the available capacity varies*

summation of the excess capacities consumes $\tilde{C}$. In this example, the value of α is set to 0.6 so that some amount of link capacity is reserved for a possible future use. The algorithm converges dynamically even when the available capacity of the link varies with the time as in Fig. 2. The residual capacity varies between from $\tilde{C} = 100$ and $\tilde{C} = 50$, which is possibly resulted by other VLs' capacity updates. The two VLs dynamically resize their capacities, and the total reservation follows the capacity changes fast. In a real situation, the changes of VL capacity may be in effect some time later due to transmission delays. The link is almost fully utilized with $\alpha = 0.8$ in this example. Note that, in the above figures, the total reservation never exceeds the link capacity in spite of the fact that the CRS handles a user's request instantaneously.

4 Multiple Link Case

In this section we discuss the extension of the capacity resizing scheme. Generally, the virtual backbone of a VPN user consists of a number of multi-hop virtual links on the provider's network. Each VL of the virtual backbone has a provisioned capacity which is allocated at the time of establishment of the VPN. As in the single link case, the VPN user may want to increase or decrease the capacity of its virtual backbone. It then generates a control packet and sends it to the CRS, which decides the new capacity for the VL and send back the control packet. In this case, however, CRS maintains the list of links that each VL traverses. Again, any request of $s_i \leq A_i$ from a VL is always accepted, and if $r_i > A_i$ after the resizing, the control packets should be generated periodically for dynamical resizing.

Let $\mathcal{V}$ be a finite set of nodes and $\mathcal{L} \subset \mathcal{V} \times \mathcal{V}$ be a set of links whose elements are unordered pairs of distinct nodes. C_ℓ is the bandwidth of a link $\ell \in \mathcal{L}$, and

R_ℓ is the summation of the excessive capacities of the VLs traversing the link. VL i has a set $\mathcal{L}_i \subset \mathcal{L}$ of links that it is associated with, and reserves capacity r_i through all links in $\mathcal{L}_i$. The set of VLs which use link ℓ can be defined by $\mathcal{K}_\ell = \{i \in \mathcal{K} \mid \ell \in \mathcal{L}_i\}$. Then, $\mathcal{K}_{\ell 1}$ and $\mathcal{K}_{\ell 2}$ are the VLs in $\mathcal{K}_\ell$ with $r_i \leq A_i$ and $r_i > A_i$ as in the single link model. We say that link ℓ is a *bottleneck link* for a VL i traversing ℓ if $C_\ell - R_\ell + f_i \leq C'_\ell - R'_\ell + f_i$ for every link ℓ' that VL i traverses; i.e., a bottleneck link of VL i is a link that has the smallest capacity available in $\mathcal{L}_i$. The CRS maintains and updates the following four information on the network status:

1. $\tilde{C}_\ell$, the available capacity of link ℓ ; initially, it is given by $\tilde{C}_\ell = C_\ell - \sum_i A_i$, $i \in \mathcal{K}_\ell$, and as the VLs update its capacities,

$$\tilde{C}_\ell = C_\ell - \sum_i A_i + \sum_{i \in \mathcal{K}_1} (A_i - r_i), \quad i \in \mathcal{K}_\ell$$

2. A_i for all $i \in \mathcal{K}$
3. R_ℓ : If we let $f_i \equiv r_i - A_i$ for $i \in \mathcal{K}_{\ell 2}$, then $R_\ell = \sum_{i \in \mathcal{K}_{\ell 2}} f_i = \sum_{i \in \mathcal{K}_{\ell 2}} (r_i - A_i)$.
4. $\mathcal{L}_i$ for all $i \in \mathcal{K}$

The admission policy of the CRS described in the previous section can be applied to the multiple link case with a small modification as follows. When the CRS receives a request from VL i , it looks up the list $\mathcal{L}_i$ and identifies a link ℓ whose $\tilde{C}_\ell$ is smaller then or equal to other links in $\mathcal{L}_i$. It is apparent that such an ℓ always exists, and CRS performs the capacity allocation at the link ℓ for the VL i as in the single link case:

Case 1 If $s_i \leq A_i$, then the request is always accepted and the information is updated by $R_\ell = R_\ell - f_i \cdot I(i \in \mathcal{K}_{\ell 2})$ and $\tilde{C}_\ell = \tilde{C}_\ell + ((A_i \wedge r_i) - s_i)$.

Case 2 If $s_i > A_i$, the decision is made according to the value of r_i as follows. If $r_i \geq A_i$, then a new capacity is assigned by

$$s_i = s_i \wedge \{A_i + \alpha(\tilde{C}_\ell - R_\ell + f_i)\}, \quad (9)$$

where $0 < \alpha < 1$. Else if $r_i < A_i$, then first increase r_i up to A_i and let $\tilde{C}_\ell = \max\{0, \tilde{C}_\ell - (A_i - r_i)\}$. Now the capacity resizing is performed by (9). $\tilde{C}_\ell$ and R_ℓ are updated according to the change of r_i . $\square$

As a VL that traverses a number of links has the same reservation r_i along the links, CRS may consider only a bottleneck link of the VL. That means the computational complexity of the allocation policy is independent of the number of links in the network or in a VL; thus, even when the size of a network grows, the work load of CRS remains the same. We simulate the policy for the example network shown in Fig. 3, which consists of four gateways connected via three links. Initially, The first and second links have the capacity of 100 and the third one 200. Starting from time 0, the capacity of the second link varies between 150 and 200, resulting the residual capacity between 50 and 100. Four VLs are established as in the figure, and the initial capacities of the VLs are provisioned

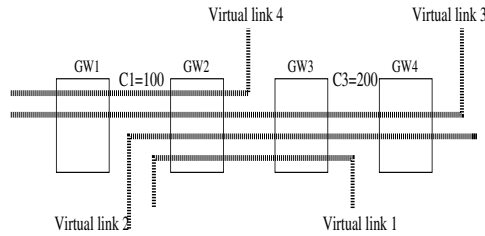


Fig. 3. Example network of four switches on which four virtual links are established.

as 20, 40, 40 and 60, respectively. Assuming that all four VLs want to increase their temporal reservations, as residual capacity of the second link varies, the three VLs which traverse the link may have chances of resizing; however, as VL 3 is constrained by the first link, its capacity can not be resized even if the second link is underutilized. Thus only the requests from VL 1 and 2 are eligible in this case. Fig. 4 shows the result of capacity resizing procedure at the example

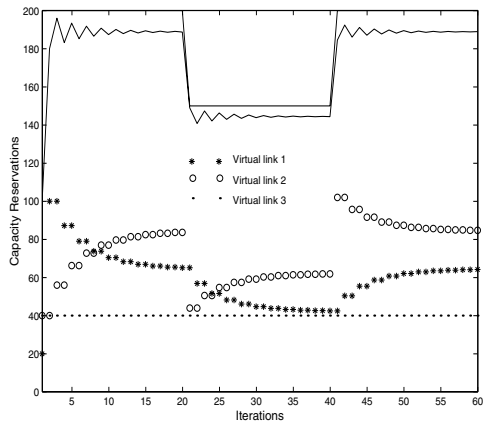


Fig. 4. Capacity resizing of three virtual links at the second link

network. Two graphs at the top depict the capacity changes of the second link: Note that this changes represent the available capacity on the link, possibly, due to the capacity resizing of other VLs which are not shown in the network. The lower part of the figure shows the capacity updates of three VLs which traverse the second link. As the residual capacity varies between 50 and 100, the capacities of VL 1 and 2 are equally fairly increased, consuming the available capacity. In this simulation, the value of α is set to 0.6. In this figure, virtual link 3 keeps sending the control packets hoping to increase its capacity, however

since its bottleneck link is the first link which has no residual capacity, CRS always sends back the requests with $s_i = r_i$ and the capacity does not change.

5 Conclusions

In this paper, we have proposed a new scheme for dynamical resizing of VL capacity based upon the users' requests. A VPN user establishes a set of VLs each of which has an initial capacity. During operation, VPN users may resize the capacity of a VL by sending a request to the CRS. They decrease the capacity possibly due to a monetary incentive, and increase the capacity to accommodate the temporal traffic increase. If the set of links along which a VL traverses have some residual capacity and thus the request is eligible, then the CRS computes a new capacity for the VL and returns back the request. We have presented that our resizing policy is simple and robust such that residual capacity is distributed fairly among competing VLs and it converges fast for a varying link capacity. Moreover, the scheme is scalable in the sense that it is independent of the size of a network and the number of hops of a VL.

Our scheme investigates the problem of fair and efficient allocation of residual link capacity among only the competing users, not the problems of how to provide a bandwidth guarantee or how to match a user's requirement. Several issues should be addressed further: The frequency of sending control packets to CRS and its impact on the performance of the network was not considered in this paper, and as the VLs may keep sending control packets even when there is no or a little residual capacity, a mechanism should be investigated such that the system becomes stable in such cases.

References

1. N. G. Duffield, *et al*, "A performance oriented service interface for virtual private networks," Nov. 1998, Work in progress, draft-duffield-vpn-qos-framework-00.txt
2. N. G. Duffield, P. Goyal, *et al*, "A flexible model for resource management in virtual private networks," in *Proceedings of ACM SIGCOMM*, NY, Mar. 1999, pp. 497–504.
3. R. Garg and H. Saran, "Fair bandwidth sharing among virtual networks: A capacity resizing approach," in *Proceedings of IEEE INFOCOM*, Mar. 2000, pp.
4. F. Bonomi and K. Fendick, "The rate-based flow control framework for the available bit rate ATM service," *IEEE Network*, pp. 25–39, Mar. 1995.
5. S. H. Rhee and T. Konstantopoulos, "A decentralized model for virtual path capacity allocation," in *Proceedings of IEEE INFOCOM*, Mar. 1999, pp. 497–504.
6. D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 1989.
7. A. Lazar, A. Orda, and D. Pendarakis, "Virtual path bandwidth allocation in multi-user networks," in *Proceedings of IEEE INFOCOM*, Mar. 1995, pp. 312–320.
8. D. Bertsekas, *Nonlinear Programming*, 1995, Athena Scientific, Belmont, MA.
9. Z. Zhang and C. Douligeris, "Convergence of synchronous and asynchronous algorithms in multiclass networks," in *Proceedings of IEEE INFOCOM*, 1991, pp. 939–943.

Throughput Improvements Using the Random Leader Technique for the Reliable Multicast Wireless LANs

Han-Chieh Chao¹, S.W. Chang², and J.L. Chen³

¹Department of Electrical Engineering, National Dong Hwa University
Hualien, Taiwan, Republic of China, 97401

hcc@mail.ndhu.edu.tw

²Southern Information Systems Inc.

Hsinchu, Taiwan, ROC

penguin@siscomm.com.tw

Department of Computer Science and Information Engineering
National Dong Hwa University, Hualien, Taiwan, ROC 97401

³lchen@mail.ndhu.edu.tw

Abstract. A new approach, random leader-based protocol (RLBP), is presented to overcome the problem of feedback collision in single channel multi-access wireless LANs for acquiring additional channels and reliability. It involves the selection of a leader in a multicast member group that acts as a representative for sending feedback to the sender using random delayed time. The leader sends nothing but acknowledgement (ACK) messages. For reliable multicasting, on erroneous reception of packets, the leader sends nothing resulting the receiver to prompt the retransmissions. For erroneous packets reception at receivers other than the leader, our protocol allows negative acknowledgements (NAK) from these receivers to collide with the acknowledgement from the leader (assuming that leader receives correct packets), thus destroying the acknowledgement that prompts the sender to retransmit the packets. Using an analytical model, it is shown that the proposed protocol obtains higher throughput than the delayed feedback-based protocol and leader-based protocol especially under lossy consideration.

1 Protocols to Avoid Feedback Collision

In multicast communications for microcell-based wireless network supporting mobile terminals, the base station communicates with a group of receivers. All communication is either directed towards the base station or directed away from the base station (base-to-terminal or base-to-terminals in the case of multicasting/broadcasting). Time is measured in terms of a basic unit called the slot. Thus, time evolves in discrete steps: 1st slot, 2nd slot, ..., nth slot, and so on. System events, like transmission/reception of a packet, occur at integer-valued slot times. It is

important to ensure that all of the entities in a cell—the base and the terminals—identify the beginning and end of slots unambiguously and simultaneously. This is the synchronization problem and for the purposes of this paper, we assumed that perfect synchronization is achieved. There are significant differences between wired and wireless LAN transmission media, which makes it impossible to use traditional wired-LAN MAC strategies like CSMA/CD to wireless LANs. In a multi-access wireless LAN, collision detection is not practical. This is because the dynamic range of the signals on the medium is very large, so that a transmitting station cannot effectively distinguish incoming weak signals from noise or its own transmission [1]. In order to prevent bandwidth loss due to collision detection (possibly due to an ACK/NAK) after the entire packet has been transmitted, a transmitter needs unambiguous and conclusive evidence that it has acquired the channel before starting a transmission. In the wireless context, this evidence can be provided by means of a handshaking mechanism implemented using short fixed-size signaling packets: Request-to-Send (RTS) and Clear-to-Send (CTS)[2][3].

We now briefly describe the RTS-CTS mechanism for unicast transmissions. When a base or a terminal transmits, it sends an RTS packet to the intended recipient. This RTS packet contains the length of the proposed transmission. If the recipient hears the RTS, it replies immediately with a CTS. The CTS also contains the length of the imminent data transmission. Upon hearing the CTS, the initiator goes ahead with the transmission. Any terminal overhearing an RTS defers all transmissions for an interval sufficient for the associated CTS to be sent and heard. Any terminal overhearing the CTS defers for the length of the oncoming data transmission. After a data packet is received, the recipient provides link-level ARQ feedback, by means of an ACK. The RTS-CTS mechanism also helps in combating the hidden terminal problem. When a transmitter about to transmit senses no carrier in its vicinity, it cannot conclude that the shared channel is unused, because another transmitter hidden from it may be transmitting at that instant. With the RTS-CTS mechanism, the hidden terminals can hear the CTS and defer using the channel. In this paper, we considered that all terminals in a cell are within range of one another and the base station. All terminals have a consistent view of what is going on in the cell and that there are no hidden terminals. A discussion on the impact of hidden terminals on our work and the means to deal with it can be found in [4]. The IEEE 802.11 Media Access Control standard uses the RTS-CTS exchange. It is important that the RTS-CTS control structure be retained when multicast functionality is overlaid. Consequently, when adding multicast functionality, we devise ways of extending the access control mechanism rather than modifying its basic structure.

While the RTS-CTS mechanism, described above, for coordination access to the channel and supplying link-level ARQ feedback works well enough for unicast transmissions, it runs into problems straight away in the context of multicasting. With the above protocol, each of the members in a multicast group would respond with a CTS to a multicast-RTS from the base, leading to a CTS collision at the base. A similar collision problem can also be expected with respect to the feedback (ACK or NAK) provided by the link-level ARQ mechanism. Standard probabilistic approaches can be used to manage the CTS collision problem. In the delayed feedback scheme,

terminals hearing a multicast-RTS send a CTS with a random delay, hoping to avoid a CTS collision. We will also consider protocols based on this idea. To tackle the ACK/NAK collision problem, a contention-based approach is possible, where receivers contend for the channel to send feedback. However, the contention-based approaches suffer from problems of their own, as will be seen in subsequent sections. This motivated us to develop a new protocol that is leader-based that satisfactorily addresses these specific problems.

1.1 Protocols

In this paper, three generic protocols are introduced., Those are delayed feedback-based protocol [5], leader-based protocol [5] and the proposed random leader-based protocol. In reliable multicasting over a multi-access wireless LAN environment, all of these protocols are for a single sender, base-station, sending reliably to a group of receivers within a cell. We assumed that the basic support for link level multicasting, such as the link level multicast address, is available at both the base-station and the receivers. The receivers that subscribe to the multicast address are considered to belong to the multicast group corresponding to that multicast address.

1.1.1 Delayed Feedback-Based Protocol [5]

In the delayed feedback-based protocol, the CTS collisions are sought to be avoided using a random timer. This protocol, termed DBP, is specified as follows:

[A] Base $\rightarrow$ Receivers (Slot 1)

1. Send multicast-RTS.
2. Start a timer (timeout period T), expecting to hear a CTS before the timer expires.

[B] Receivers $\rightarrow$ Base

1. On hearing RTS, start timer with an initial value chosen randomly from $\{1, 2, \dots, L\}$.
2. Decrement timer by 1 in each slot.
3. If a CTS is heard before timer expires, freeze timer (CTS suppression).

If no CTS is heard before timer expires, send CTS.

[C] Base $\rightarrow$ Receivers

If no CTS is heard within T , back off and go to Step A.

If a CTS is heard within T (at a random time), start data transmission.

After completing transmission, prepare to transmit the next packet and go to Step A (no waiting for feedback).

The next step is executed only when a multicast transmission occurs in step C.

[D] Receivers $\rightarrow$ Base

If a packet is received without error, do nothing.

If an error occurs, contend for the channel to send NAK.

1.1.2 Leader-Based Protocol [5]

The Leader-based protocol assumes that one of the receivers of the multicast has been chosen to be a leader for the purpose of supplying CTS and ACK with a response to RTS and data packets of length $\mathcal{L}$ respectively. The leader-based error recovery protocol (LBP) is specified as follows:

[A] Base $\rightarrow$ Receivers (Slot 1)

Send multicast-RTS.

[B] Receivers $\rightarrow$ Base (Slot 2)

Leader: If ready to receive data, send CTS.

If not ready to receive data (e.g., due to insufficient buffers), do nothing.

Others: If ready to receive data, do nothing.

If not ready to receive data, send NCTS (not clear to Send).

[C] Base $\rightarrow$ Receivers (Slot 3)

If a CTS was heard in slot 2, start multicast transmission.

If no CTS was heard in slot 2, back off and go to Step A,

The next step is executed only when a multicast transmission occurs in Step C.

[D] Receivers $\rightarrow$ Base (Slot $(\mathcal{L}+3)$)

Leader: If a packet is received without error, send ACK.

If an error occurs, send NAK.

Others: If a packet is received without error, do nothing.

If in error, send NAK.

LBP uses both ACKs and NAKs from receivers as feedback to the sender. It makes an interesting use of collisions associated with one or more NAKs to ensure that the sender does not get a positive feedback if one or more group members received an erroneous transmission.

1.1.3 Random Leader-Based Protocol

We now provide our random leader-based protocol for reliable multicasting over a multi-access wireless LAN. Both ACKs and NAKs are used to provide reliable transmissions. When the first multicast-RTS packet is sent from the base station, the DBP method is used to get the CTS packet from any of the member receivers. The timer for that receiver is then set to 0. The receiver then becomes a pseudo leader or named as random leader.

The RLBP is specified as follows:

[A] Base $\rightarrow$ Receivers (Slot 1)

1. Send multicast-RTS.
2. Start a timer (timeout period T), expecting to hear a CTS before the timer expires.

[B] Receivers $\rightarrow$ Base

1. On hearing RTS, start the timer with an initial value chosen randomly from $\{1, 2, \dots, \mathcal{L}\}$.
2. Decrement timer by 1 in each slot.
3. If a CTS is heard before the timer expires, freeze the timer (CTS suppression).

If no CTS is heard before the timer expires, send CTS.

4. If a CTS is sent, set the timer=0 when the next packet comes.

[C] Base  **Receivers**

If no CTS is heard within T, back off and go to Step A.

If a CTS is heard within T (at a random time), start the data transmission.

After completing the transmission, prepare to transmit the next packet and go to Step A (no waiting for feedback).

The next step is executed only when a multicast transmission occurs in step C.

[D] Receivers  **Base**

If the packet is received without error, send ACK.

If an error occurs, contend for a channel to send NAK.

Compared to the LBP, the base in the RLBP scheme does not need a complex mechanism to maintain the leader table or refresh it whenever the leader leaves.

1.2 Discussion

In comparison to the LBP, a successful RTS-CTS exchange would take longer in both DBP and RLBP. This is because DBP has to deal with the possibility of CTS collisions, as well as RLBP. RLBP takes even longer time for the first packet of a multicast transmission. As DBP and RLPB are NAK-based, a packet must be maintained longer to ensure that most of the retransmission requests can be serviced. At the receiver, a greater buffer is required to buffer out-of-order packets so that the upper layers receive an ordered delivery. Another problem with DBP is the choice of the right parameters for waiting times and the feedback probability. This choice is dependent upon the number of group members. The group members are not likely to have an estimate of the group size. It is possible for the sender to do this estimation and send out the right parameters with the RTS to prevent complex estimation mechanism implementation. A difficulty arises if the leave-group message sent by the leader, leaving group G_i , is not heard at the base station. The base station will wrongly believe that G_i has a leader although the leader has already signed off. In this case, when the base sends out a multicast-RTS for group G_i , it will hear no CTS. After several unsuccessful attempts, the base will erase the leader entry corresponding to G_i , and stop forwarding packets addressed to this group. The LBP and RLBP, in a loss-free channel, have similar performances. When we consider the packet loss situation, the throughput will be different. We will discuss this in the next section.

2 Performance Study

In this Section, we compare the performance of LBP, DBP and RLBP. We consider a scenario where multicast traffic is the only traffic present in the cell and whether the control packets will be lost. Time is measured in terms of a basic unit called the slot.

The basic criterion used for studying the performances of these three protocols is the channel holding time associated with a tagged data packet. We can take a measure of the throughput using this criterion.

We consider two cases in this chapter. The first is error-free and the second is the retransmission requirement. The error-free channel is an idealized case, since all packets are received correct. In the second case, we derive a lossy model to illustrate the retransmission requirements.

2.1 Error-Free Transmission

In DBP, a receiver hearing a multicast-RTS from the base starts a timer with a value chosen at random from the set $\{1, 2, \dots, L\}$. We assume that the value L is made available to the receivers by this case; for example, it may be carried in a field in the RTS packet. The receiver whose timer expires sends a CTS. Upon hearing the CTS, the other receivers, whose timers have not yet expired, suppress their own CTSs. A CTS collision occurs if two or more receivers happen to choose the same initial value for their timers.

Since the receivers send the CTS after a delay, the base must wait for some time to hear the CTS. This is the base's timeout period for T slots. If a base does not hear the CTS within time T , it assumes there was a collision, and tries again. We choose $T < L$. This is because if T is large, then a lot of time is wasted before the base times out. On the other hand, choosing a moderately large L helps in avoiding a CTS collision within T .

If we give the number of receivers N , L and T , the probability that the base hears a CTS within time T , called p_h , can be expressed as follows;

$$p_h = \frac{N}{T} \sum_{i=1}^T \left(\frac{L-i}{L} \right)^{N-1}$$

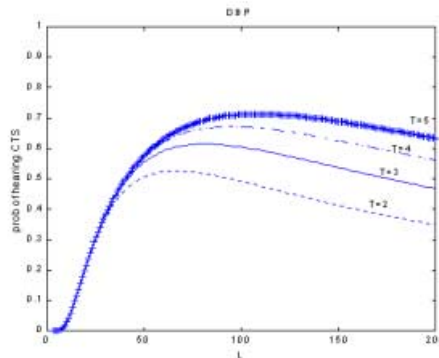


Fig. 1. Variation of p_h with L , keeping N and T fixed.

In Figure 1, we show how p_h varies with L , when N and T are fixed. In all cases we find that p_h first increases, hits a peak and then decreases as L is increased. When L is small, the chances for CTS collision increase. When L is large, the chances for no receiver to send a CTS within the timeout period T increased. The best values for p_h are therefore found in the middle.

In order to create a situation favorable to DBP, the following assumption is made:

Assumption S: If no CTS is heard within the timeout period T , the base does not back off.

Under this condition, we ask the question: on the average, how long does the base spend in the access period?

Let T_a^{DBP} be the random variable representing the total time spent by the base in the access period, measured from the instant when it is ready to send the first RTS. We assume that it takes 1 slot to transmit the RTS or any other control packet. Let A denote the event that the base hears a CTS within T slots of sending the first RTS, and $\bar{A}$ denotes the complementary event. Then we have

$$T_a^{DBP} = \begin{cases} 1 + \tau & \text{if } A \text{ occurs} \\ (1 + T) + W_a & \text{if } \bar{A} \text{ occurs} \end{cases}$$

when $\tau \leq T$ is the time at which the CTS is heard, if A occurs, and, W_a is the time spent in the access period after the first timeout.

Now the distribution of W_a is the same as the distribution of T_a^{DBP} , and according to $\text{Prob}(A) = p_h$, we obtain

$$E(T_a^{DBP}) = E(\tau / A) + \frac{(1 - p_h)}{p_h} T + \frac{1}{p_h}$$

In figure 2, we presented some examples of how $E(T_a^{DBP})$ varies with the parameters L and T . The number of receivers, N , is chosen to be 30. For a fixed T , $E(T_a^{DBP})$ first decreases, reaches a minimum and then increases again as L is increased. This is because $E(T_a^{DBP})$ is high when p_h is low and vice versa.

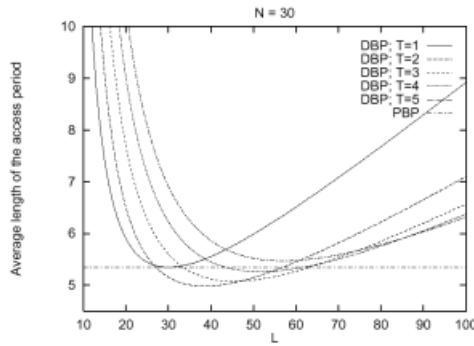


Fig. 2. Comparison of the expected time of DBP [4]

In the roaming scheme, all members may roam to another cell using a random probability. Since every CTSs is sent using a random delay mechanism whenever multicast-RTS comes, it is not necessary to discuss the roaming probability.

In the LBP protocol, after hearing the multicast-RTS, a receiver sends a CTS in the next slot with probability p . The base waits for 1 slot after sending the RTS. If exactly 1 member happens to reply, the access period is complete. If the base does not hear a CTS, it then has to restart the process by sending the multicast-RTS again.

The minimum time spent in the access period is 2 slots, 1 to send the RTS and 1 to hear the CTS. Let p_0 be the probability that the access period lasts 2 slots. Then we can get

$$p_0 = Np(1-p)^{N-1}$$

under Assumption S, the number of attempts necessary for the access period to be complete is geometrically distributed with parameter p_0 . Hence the mean time spent in the access period, $E(T_a^{LBP})$, is given by $2/p_0$. To minimize this time we choose p so that p_0 is maximized. This is achieved for $p=1/N$, giving the following expression for the mean time:

$$E(T_a^{LBP}) = \frac{2}{(1-1/N)^{N-1}}$$

The N value can be transmitted to the receivers from the base in a field of RTS packets, for example.

Considering the roaming scheme, when the leader roams to another cell, the base station must choose one of the other members as a leader. This mechanism requires 2 slots. Assume that the roaming probability is p_r , and the mean time for LBP is:

$$E(T_a^{LBP}) = \frac{2}{(1-1/N)^{N-1}} + 2p_r$$

Finally, we consider RLBP protocol. Although there is a fake leader, the RLBP leader was chosen using a random delay mechanism. If there is no leader, the mean time spent in the access period, $E_d(T_a^{RLBP})$, is equal to the mean DBP cost time. If there is a fake leader, the mean time spent in the access period $E_i(T_a^{RLBP})$ is equal to the mean LBP cost time. Considering the roaming scheme, if the fake leader stays in the same cell, when the base station sends a multicast-RTS, the fake leader will immediately send a prepare to receive data message. The other members will be quiet after hearing a CTS. If the fake leader roams to another cell, there is no longer a leader. All of the members will start a random back mechanism to send a CTS reply. The receiver which sent a CTS will become the fake leader and the delay timer will be set to 0.

Assume that the roaming probability is p_r , the mean time cost of RLBP is:

$$E(T_a^{RLBP}) = (1 - p_r)E(T_a^{LBP}) + E(T_a^{DBP})p_r$$

Hence,

$$E(T_a^{RLBP}) = (1 - p_r)\left(\frac{2}{(1 - 1/N)^{N-1}}\right) + p_r\left(E(\tau/A) + \frac{(1 - p_h)}{p_h}T + \frac{1}{p_h}\right) \quad \text{Cost Under DBP, LBP and RLBP}$$

Consider DBP, when the channel is error-free, no NAKs are necessary because no packet is received in error. The base station then transmits a multicast-RTS and waits for the timeout period T to hear a CTS. After several possible attempts, the base hears the CTS and transmits the packet.

We focused on a tagged packet and considered the mean time required to transmit a packet. Including the time spent in the access period. We consider this time to be the cost associated with the tagged packet. The cost to transmit a packet gives a measure of the efficiency of the protocol. Let the data packet transmission time be C slots. The cost of a packet under DBP is then $E(T_a^{DBP} + C)$.

Consider the events on the channel under LBP, a packet transmission is preceded by 2 slots: 1 for the multicast-RTS, immediately followed by an ACK packet which also occupies 1 slot. Thus, the cost of a packet transmission under LBP is : $(C+3+2p_r)$.

The events on the channel under RLBP, when fake leader roaming occurs, the RLBP cost is equal to the DBP cost, otherwise the cost is equal to the LBP cost. Assume that the roaming probability is p_r , the cost of a packet transmission under RLBP is : $(1-p_r)(C+3)+p_r(E T_a^{DBP})$.

Assume that it takes 20 slots to transmit a data packet ($C=20$), and drive the roaming probability at 0% and 40%, we obtained figures as follows (Figure 3(a)-(b)).

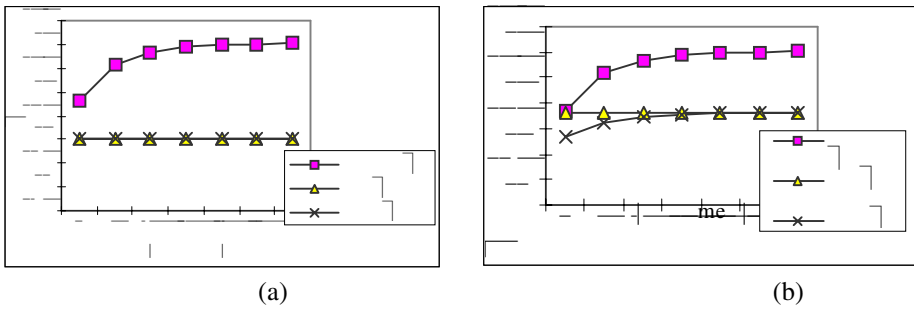


Fig. 3. (a) Cost under roaming probability 0%, (b) Cost under roaming probability 40%

From figures 3-4, we can see that the cost of LBP and RLBP are very close in an error-free channel but better than the best performance achieved by DBP.

2.2 Lossy Channel

When the channel is lossy, packets are received in error and retransmissions are required. In a lossy channel, control packets like RTS, CTS, ACK, NAK may be lost because of collision or other problems. In DBP, since the CTS packets are not sent by any particular receiver, the mean time cost will not be changed in DBP. In LBP, when the leave message sent by the leader is lost, the base station will try several times with multicast-RTS to prepare to send data. After several unsuccessful attempts, the base will erase the leader entry corresponding to that group, and stop forwarding packets addressed to that group. If there are other group members that are still interested in this group, they will eventually time out and start the subscription process for the group again. This mechanism may require much time to produce a new leader.

The RLBP protocol does not maintain a group-leader table at the base. The base will not expect a CTS packet from a particular receiver. When a fake leader leaves without acknowledgement, the other receivers will not hear a CTS packet from one another. A random delay process will then begin to decide a new fake leader. When packet loss occurs, the mean time cost will equal DBP. When no packet loss occurs, it will be equal to LBP.

We set the loss probability (LP) as 5% and 10%, and cost under roaming probability (CRP) as 10% and 40%. We obtained the following figures. (Figure 4(a)-(d))

From these figures, we can see that the LBP and DBP are close under high roaming, lossy models. This is because when packets are lost, the LBP protocol must spend more time to rebuild the multicast tree. The greater the loss probability, the greater the time spent. When the number of members becomes larger, the time spent in rebuilding the multicast tree will increase. We can see that the RLBP has more efficiency when the roaming probability and loss probability become higher. This is because the RLBP protocol does not need to rebuild a multicast tree if the base

receives no CTS messages for a long time. It will automatically pick another one in the group as the leader.

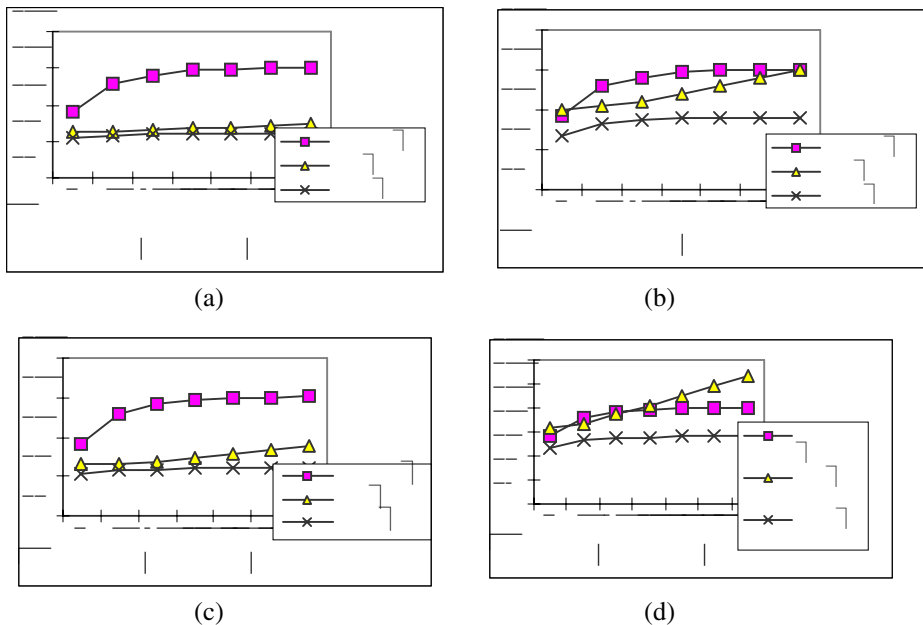


Fig. 4. (a) CRP = 10% LP = 5% (b) CRP = 40% LP = 5% (c) CRP = 10% LP = 10% (d) CRP = 40% LP = 10%

3 Conclusions

In this paper, the proposed protocol allows responses from a pseudo leader to avoid collision possibilities, and decrease the problems caused by packet loss. The base station does not need much time to wait for CTS messages. The RLBP protocol provides very efficient solution to the CTS and ACK/NAK collision problem. Compared to the LBP protocol, RLBP does not waste time when control messages are lost and it is not necessary to choose a receiver to transmit the CTS with a random delay mechanism. In addition, this method is very simple to implement and can be integrated easily into the current wireless LAN standard. A group-leader table is not maintained at the base, so the base will not expect a CTS packet from a particular receiver. For the LBP and RLBP methods in a loss-free channel, the performances of these two protocols are similar. Under packet loss environments, the RLBP protocol is prevailing.

References

- [1] W. Stallings, "Local and Metropolitan Area Networks." 5th Edition, Prentice Hall, 1997.
- [2] Wireless Medium Access Control and Physical Layer Working Group", 802.11 IEEE Draft Standard-Wireless LAN. IEEE Standards Department, D3, Jan 1996.
- [3] P. Karn, "MACA—A New Channel Access Method for Packet Radio." ARRL/CRRL Amateur Radio 9th omputer Networking Conference, Sep 1990.
- [4] Joy Kuri and Sneha K. Kasera "Reliable Multicast Multi-access Wireless LANs." Center for Electronic Design and Technology (CEDT) Tech Report, Indian Institute of Science, Jan, 1999.
- [5] Joy Kuri and Sneha K. Kasera "Reliable Multicast in Multi-access Wireless LANs." Proceedings of IEEE Infocom 99, , Volume: 2 , Page(s): 760 –767, New York, USA, March 21-25, 1999.

Performance Evaluation and Implementation of QoS Support in an 802.11b Wireless LAN

Vinod Mirchandani and Eryk Dutkiewicz

Communications and Networking Laboratory
Motorola Australian Research Center
Locked Bag 5028, Botany NSW 1455
Australia
{vinm, eryk}@arc.corp.mot.com

Abstract. The availability of wireless technologies such as HomeRF, WaveLAN and Bluetooth have propelled the demand for their use in the office, home and public spaces. These technologies can provide wireless multimedia services. QoS for these services can be provided by means of a MAC level QoS or a network level QoS. In this paper we describe the implementation of a network level QoS for the 802.11b technology by using the Windows 2000 Operating System. The testbed is used to obtain throughput of UDP flows created by applications such as iperf and qtcp for different number of active stations, packet sizes and packet loss. These performance evaluation results are validated by means of stochastic simulation of the testbed by using GloMoSim.

1 Introduction

The increase in computer processing power and availability of high-bandwidth communication networks has fuelled the demand for networked multi-media desktop communications. Users of broadcast television and telephone networks have the same level of expectancy for Quality of Service (QoS) from networked multimedia services. In order to meet such quality of service expectations the IETF has been developing a number of mechanisms for quality of service delivery. These include IntServ and DiffServ as well as associated protocols RSVP, RTP and RTCP [1-2]. There is currently a lot of thrust in the information technology industry to provide services using wireless networks, exemplified by the arrival of new and more cost-effective technologies such as WaveLAN, HomeRF and Bluetooth. These technologies provide mainly asynchronous mechanisms for data delivery, although some also provide isochronous channels for delivery of time sensitive services.

Wireless technology based on the 802.11 standard is currently a popular choice for data delivery in wireless LAN environments. Although, the standard allows for support of isochronous channels, in practice this has not been widely implemented. There is also some activity to extend the current standard to support QoS services by extending the capabilities of the MAC layer [3]. However, such extensions will increase the complexity of the MAC layer and this in turn may increase the cost of the technology. Rather than providing QoS support at the MAC layer, a more robust and effective approach may be to provide such support at the network layer. Such an

approach would not depend on the existence of a QoS-aware MAC layer and could also be applied to existing wireless LAN technologies. A number of operating system developers have provided QoS features in their operating systems. Microsoft, for example, has provided extensive QoS support in its Windows 2000 operating system and limited QoS support in Windows 98 operating system. The QoS functionality provided by Microsoft is based on the RSVP signaling protocol that conveys the reservation requests made by the receiver to the admission control server. Some work in the provision of QoS mechanisms is also being done in Linux [4-5] and free BSD [6]. However, at the moment these operating systems are not completely developed from the QoS perspective. In addition, their QoS functionality can be accessed only at the kernel level. Based on the availability and functionality of QoS features in various operating systems we have chosen Windows 2000 as the basis for the implementation and testing of network-level QoS support in our wireless LAN testbed.

In this paper, we describe our efforts in the implementation of a network-level QoS mechanism in a wireless LAN testbed based on the 802.11b technology. The performance of the testbed is evaluated by means of simulations and testbed measurements. The results obtained from these tests are then used to evaluate the efficacy of network-level QoS for providing QoS support in shared medium wireless networks. Section 2 presents an overview of the wireless LAN testbed. Section 3 shows our simulation and testbed measurement results. Section 4 describes the testing of the suitability of network-level QoS support. Section 5 presents our conclusions for this work.

2 Wireless LAN Testbed Overview

Figure 1 shows a high-level representation of the wireless LAN testbed.

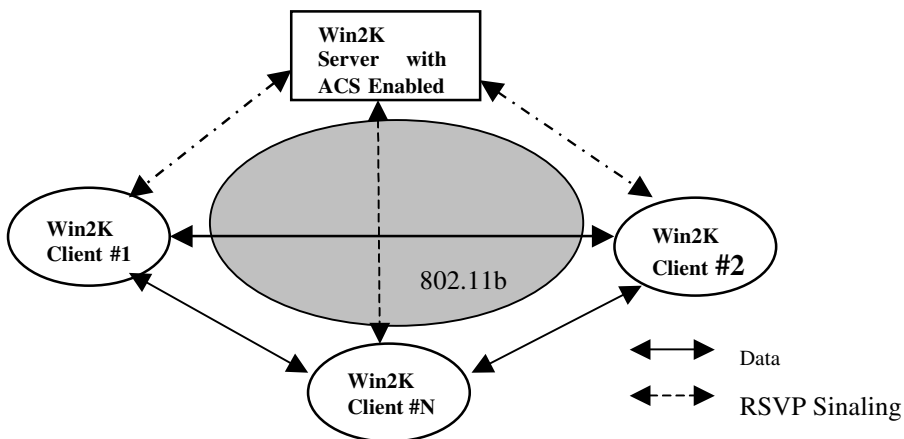


Fig. 1. Topology of the Wireless LAN Testbed

The testbed consists of a number of Windows 2000 professional operating system based clients and a Windows 2000 server operating system based server. The PC machines in the testbed are connected wirelessly using ORNICO cards which use the 802.11b protocol for medium access. The ORNICO cards are configured to operate in the adhoc mode allowing more flexibility in topology configuration.

TCP/IP and packet scheduler modules must be installed on the clients to enable their QoS functionality. The purpose of the packet scheduler is to schedule the packets to meet the flow requirements. The Admission Control Service (ACS) installed in the server offers logical reservation of the resources by consulting the policies entered by the network administrator. The ACS server stores the flow policies for each of the flows to be created between the sender and receiver. The policy setup mechanism offers granularity in the setup by means of a hierarchical QoS policy setup mechanism.

3 Performance Evaluation of Wireless LAN

The main aim of our performance evaluation studies was to determine the limitations of the wireless LAN to support real-time IP streaming applications. In this study we have concentrated on simple applications which do not have the ability to control its transmission rate in response to network conditions. Due to small MAC buffer sizes in the wireless LAN and the lack of rate control at the source; packet loss is expected to be the critical measure of quality of service for such applications.

3.1 Simulation Set-Up

The simulation model of the wireless testbed is shown in Figure 2.

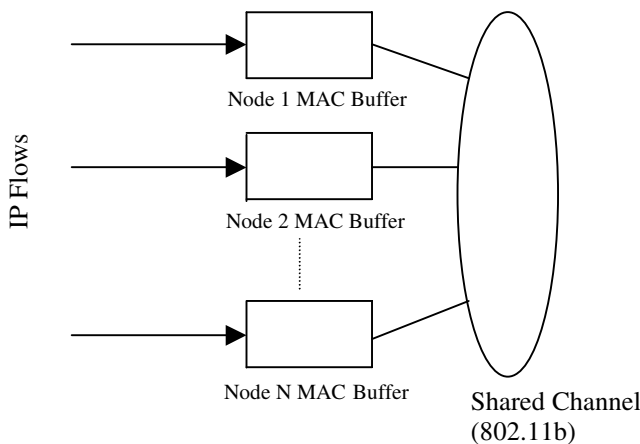


Fig. 2. Simulation Model for Capacity Limit Determination in Wireless Testbed

In order to determine capacity limits of the wireless testbed we simulated it using the GloMoSim package developed by the University of California, LA. We used the freely available GloMoSim Version 1.2.3 [7]. Each node can transmit a flow of IP packets. IP packets are buffered in MAC layer buffers while contending for access to the shared wireless channel. In all our tests we used UDP flows. UDP packets may be lost due to the MAC buffer overflow during periods of channel contention. In the simulation model we were interested in determining the maximum allowable transmission rate from each node as a function of active nodes given a particular QoS constraint in terms of packet loss.

Our wireless network simulations used the following base parameters:

- channel capacity rate = 11Mbps
- DSSS physical layer
- CBR UDP flows with fixed packet payload sizes
- RTS/CTS option disabled
- MAC queue size = 25 frames

3.2 Capacity Limits with QoS Constraints

Figure 3 shows the 3D admission region for 1% packet loss.

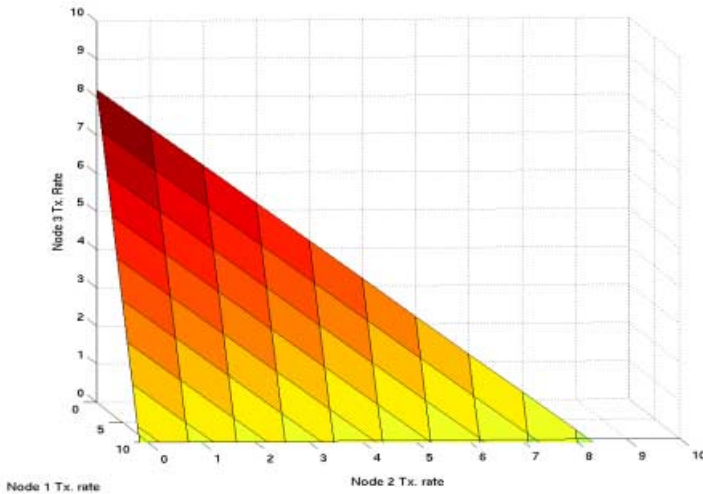


Fig. 3. Maximum Allowable Transmission Rates in Mbps for 1% Level of Packet Loss Constraint.

Table 1 shows the maximum available transmission rates for the case of 3 nodes using 3 different IP packet sizes. Each entry in the table shows the sum of the transmission rates from the 3 nodes assuming that the 3 nodes are transmitting IP packets at an equal rate. This is expected to be a conservative capacity measure due to the higher level of channel contention compared to scenarios in which transmitting rates are not equal. The maximum packet size of 2000 bytes was chosen to be below the threshold for 802.11 MAC frames in order to avoid IP packet fragmentation.

Table 1: Maximum Available Transmission Rates for 3 Active Nodes as a Function of Packet Payload Sizes for Two Different Packet Loss Constraints

IP packet Size (bytes)	1% packet loss constraint	10% packet loss constraint
512	4.470 Mbps	4.920 Mbps
1460	7.400 Mbps	7.800 Mbps
2000	7.800 Mbps	8.550 Mbps

The simulation results in Table 1 show the expected behavior: the maximum available transmission rates increases as the packet loss constraint becomes less stringent and as the packet size increases.

The following recommendations are made based on the simulation study :

- The above simulation measurements have been obtained for fixed size packets but actual applications will generate traffic with variable packet sizes. Therefore, we need to obtain capacity limits given particular packet size distributions.
- Admission control policies should take into account capacity limitations.

3.3 Testbed Measurements

The wireless testbed of Figure 1 was implemented and capacity measurements were obtained from it. The main motivation for obtaining the measurements from the testbed was to validate the trend observed from the simulation results given in table 1.

Two tools for measuring capacity limits in a WLAN which we have investigated were Qtcp from Microsoft and Iperf from [8]. Both these tools provide a means to generate UDP packets. However, they cannot provide packet loss measurements since they use a blocking mechanism which reduces the UDP transmission rate to avoid packet loss. As such using these tools can only give approximate results of allowable transmission rates.

In our tests we used iperf since it has a better user interface. The tests using 1460 byte long packets indicated that the maximum capacity of the WLAN is around 6 Mbps With 512 byte long packets this capacity was reduced to around 3.5Mbps. These rates are well below those obtained from our simulations. The discrepancy can be attributed to the fact that the simulation model assumed an error-free environment and a fixed transmission rate of 11 Mbps from each node. On the other hand, an auto transmission rate mechanism is implemented in the Lucent driver and it most likely adjusts the transmission rate between 5.5Mbps and 11Mbps, depending on the prevalent radio interface conditions.

4 QoS Support Testing in Wireless LAN

In order to confirm the working of the QoS mechanisms in our testbed it was necessary to use QoS enabled applications. Although, the Windows 2000 operating system has QoS capabilities to our knowledge there are currently only two applications that are QoS enabled: Qtcp and Netmeeting. Qtcp is a command driven

utility whose main objective is to determine the end-to-end packet delay and packet loss in the test network. The utility allows the user to generate a stream of packets of different traffic types, packet sizes and transmission rates. It also allows the user the choice using RSVP's reservation signaling. Correct operation of the QoS mechanism was verified for the Netmeeting application. The Netmeeting application produces three streams – RSVP signaling, a video stream and an audio stream. The video stream is of the controlled load service level, audio stream conforms to the guaranteed service level and the RSVP signaling is done by using the best effort service level. The QoS policies were set in the Admission Control Server (ACS).

The ACS plays a pivotal role for the QoS provisioning in the Windows 2000 OS. The QoS ACS is configured with policies that reflect the QoS requirements of the enterprise, with additional policies created for groups or individual users. The QoS ACS must be the member of the same domain as the subnet it intends to manage. This involved first creating a managed subnet and assigning the ACS managing the subnet. Then the QoS ACS policies for the subnetwork constituting the testbed were set. After this the policies specific for individual nodes were set. The policies are spread over two general parameters – Flow limits and Aggregate Limits. The flow limits and aggregate flow limits encompasses data rate, peak data rate and duration of the flow parameters.

The QoS ACS policy was set to offer unlimited bandwidth, peak data and unlimited resource to the Netmeeting application. The messages in the ACS log and the RSVP log files of the ACS confirmed that the reservation requests messages made by the receiving entities in both the clients had succeeded. When the bandwidth parameters in the policy for the clients was decreased to limit the flow of Netmeeting streams to 5kb/s the traffic monitor utility indicated that the Netmeeting audio and video streams had changed to best effort service level. Similar behavior as the Netmeeting application was observed for the Qtcp application which confirmed that the Qtcp was a QoS enabled application.

5 Conclusion

We have implemented and demonstrated the operation of a QoS mechanism on a wireless testbed that used the Windows 2000 OS Professional and server based machines. A novel method based on table look up procedure is proposed for setting up the policies dynamically in the admission control server. The look up table was obtained by using GlomoSim simulation tool. The practical results show that the WaveLAN can have a maximum throughput of around 6 Mbps and this affects the accuracy for validating the simulation results. Even then the results obtained by both simulation and experiment follow similar trends for different packet sizes and packet loss rate. The Wavelan transmission rate cannot be set by the Lucent driver which is a limitation of the driver. Above all the proposed method would facilitate in a quick set up of policies in the admission controller for a given number of active nodes.

References

1. Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification. R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin. RFC 2205, September 1997.
2. RTP: A Transport Protocol for Real-Time Applications. Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. RFC 1889, January 1996.
3. <http://grouper.ieee.org/groups/802/11/Documents/>, IEEE 802.11 Documentation.
4. <http://www.cs.umass.edu/~lass/software/qlinux>, QLinux: A QoS enhanced Linux Kernel for Multimedia Computing .
5. http://www.ittc.ukans.edu/~pramodh/courses/linux_qos/qos_linux.html QoS support in Linux
6. <http://www.freebsd.org>, FreeBSD.
7. <http://pcl.cs.ucla.edu/projects/glomosim>, GloMoSim Global Mobile Information Systems Library.
8. <http://dast.nlanr.net/Projects/Iperf>, Iperf version 1.1.1, Feb 2000.

Increasing Throughput and QoS in a HIPERLAN/2 System with Co-channel Interference

Jürgen Rapp

Communication Networks, Aachen University of Technology,
Kopernikusstr. 16, D-52074 Aachen, Germany
rapp@comnets.rwth-aachen.de,
<http://www.comnets.rwth-aachen.de>

Abstract. There is a growing demand for bandwidth as well as mobility. Within ETSI BRAN a wireless LAN called HIPERLAN/2 has been standardized. While data rates can be as high as 54 Mbit/s for a high carrier to interferer ratio (C/I), more robust combinations of modulation and code-rate have to be used and also retransmissions do occur when interference is present. This leads to much smaller effective data rates. Interference and link adaptation are therefore important topics. In order to have a realistic co-channel interference, two radio cells are implemented which interact with each other. In both radio cells detailed implementations of the protocols are used. Data transmission between the terminals is carried out via TCP/IP respectively UDP. In this paper it is shown how throughput as well as delay can be improved in all load conditions by reducing co-channel interference and reducing the variations in the interference situation which significantly increases the effectiveness of link adaptation.

1 Introduction

There is a growing demand for bandwidth as well as mobility which led to several research projects which are investigating high speed wireless LANs. Within ETSI BRAN a wireless LAN called HIPERLAN/2 has been standardized which operates in the 5–6 GHz band. It can be used in combination with e.g. ATM or TCP/IP and as part of UMTS.

Within HIPERLAN/2 the modulation and code-rate are adapted to the conditions of the radio link. With seven different combinations of modulation and code-rate (=Phy. Mode), data rates from 6 Mbit/s to 54 Mbit/s with different requirements on the C/I and different resilience against transmission errors are possible [1]. HIPERLAN/2 is a cellular system with frames of a fixed length of 2 ms. A frame starts with a Broadcast Channel (BCH) followed by a downlink (DL)- and uplink (UL)- phase and the Random Channel (RCH). HIPERLAN/2 uses Time Division Duplex (TDD) and Time Division Multiple Access (TDMA). In the centralized mode an Access Point (AP) serves as central controller. It decides every frame anew when a wireless terminal shall receive and when it shall transmit and which Phy. Modes shall be used for transmission.

2 Goal of this Work

While data rates can be as high as 54 Mbit/s for high C/I values ($>30\text{dB}$), more robust Phy. Modes have to be used when interference is present. Also more re-transmissions do occur if interference becomes higher. This leads to much smaller effective data rates. Therefore, interference and link adaptation are important topics. In this paper several approaches are presented which can improve the interference situation and which can make link adaptation more effective. First, one approach is described which works under low and medium load conditions. Then it is shown how this approach can be used to improve the situation especially for delay sensitive connections. Finally, an approach is presented which works under all load conditions. It improves system performance if the first mentioned approaches are not used and increases system performance even more if the above mentioned approaches are used.

2.1 Improving the Interference Situation and Effectiveness of Link Adaptation in Low and Medium Load Conditions

In low and medium load conditions not all the frame is used for transmission. These silent periods can be exploited to reduce the interference in co-channel radio cells and to make link adaptation easier. Unfortunately, most traffic in LANs is data traffic which is transmitted via TCP/IP. Investigations have shown that this traffic is quite bursty in nature, especially considering WWW or FTP traffic [2]. The burstiness of the data leads to strong variations in the lengths of the silent periods. Thus, in low and medium load conditions, there are many frames which are completely filled and also many frames which are almost empty. The strong variations of the silent periods mean also strong variations in the interference situation for the co-channel radio cells which makes link adaptation difficult. For link adaptation it is fortunate if the interference situation that is measured now equals the interference in the future. Link adaptation works best for a slowly changing and predictable interference situation.

In [3] it has been shown that the burstiness in the interference situation is reduced with an approach called Reduced Burstiness (RB). Even if big TCP segments arrive at the AP, the scheduler does not permit one terminal to use the whole frame even if the current system load would allow this. Instead every terminal is allowed to use a certain percentage of the frame at maximum. This percentage depends on the number of active terminals and their expected mean load. It is chosen in a way that no terminal can use the whole frame by its own but that all active terminals together can fill the whole frame.

By limiting the percentage of a frame that can be used by one terminal the burstiness of transmission is reduced. This is shown in Fig. 1 where the duration of the silent periods is shown for RB and the approach that does not limit the burstiness of transmission (Normal Burstiness (NB)).

With RB, the duration of the silent periods is almost equally distributed between 0 ms and the maximum length of the silent periods (see Fig. 1). Since BCH and RCH are present in every frame, the duration of the silent period can

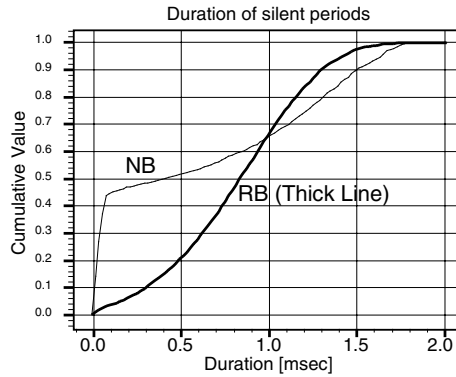


Fig. 1. CDF of duration of silent periods

not become longer than 1.83 ms in this scenario. The standard deviation for the duration of the silent periods is reduced from 0.62 ms to only 0.38 ms. These results were obtained for a load that led to a mean frame usage of 60% which means that in average 60% of the frame were used.

In [3] it has also been shown that with RB the retransmission load is significantly reduced compared to NB. Thus, RB achieves a higher throughput for a given delay requirement or a smaller delay for a given throughput.

Furthermore, it was shown that an intelligent placement of the silent periods is necessary to increase throughput and QoS. Several placements of the silent period were investigated (see Fig. 2):

- The silent period is divided by the number of scheduled terminals. An equal duration of the silent period is inserted before every scheduled PDU burst (Equal Silent Period Placement (*ESP*)).
- The silent period is inserted as one piece before the BCH in both co-channel radio cells (Symmetric Silent Period Placement (*SSP*)).
- The silent period is inserted as one piece before the BCH in one radio cell and after the BCH in the co-channel radio cell (Asymmetric Silent Period Placement (*ASP*)).

Since the position of the silent periods can be chosen without any restrictions on a frame per frame basis, the above describe approach works for all offsets which can occur between the frames of two co-channel radio cells [3].

2.2 Improving the Interference Situation for Delay Critical Connections in Low and Medium Load Conditions

In this paper it will be shown that with RB and an intelligent placement of the silent periods it is possible to improve the situation especially for delay sensitive connections which must not suffer any or many retransmissions. This can be done

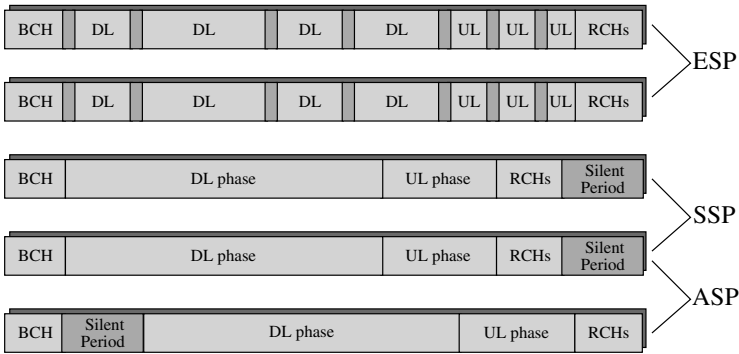


Fig. 2. Different placements of the silent periods

by the above described approach and scheduling delay sensitive connections in parts of the frame where interference will be usually lower than in other parts of the frame. In order to reduce the interference in certain areas of up- and downlink in *both* radio cells, a different version of ASP was used. It is called ASP-Both Links Improved (ASP-BLI) and its structure is shown in Fig. 3. The most delay critical connections can be transmitted in the areas where interference will be usually lower than in other areas.

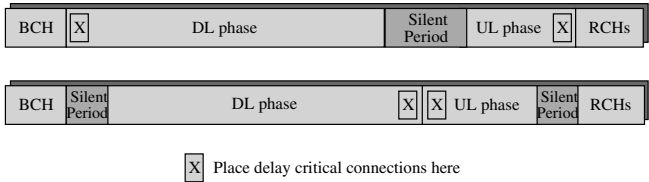


Fig. 3. Placement of the silent periods with ASP-BLI

Please notice that the length of the unused capacity, the length of the downlink and uplink phase and the length of the BCH varies from frame to frame. Nevertheless it will be shown that with this approach it is possible to significantly improve the situation for delay critical connections in *both* radio cells and for *both* up- and downlink.

2.3 Further Improvement of the Interference Situation and Effectiveness of Link Adaptation for All Approaches in All Load Conditions

The above described approaches work well under low and medium load conditions. For higher load values their effect becomes very small since they depend on the existence of unused capacity in the frame. Now another approach will

be presented which works under all load conditions. It can be used in combination with the above described approaches or without them. If it is used together with the above described approaches it leads to a further improvement of system performance.

In [3] it was shown that the effectiveness of LA is increased if the interference situation becomes less varying. This is also tried to achieve with the following approach. It is called Maximum Similarity (MSI) and tries to achieve as much likeliness between consecutive frames in a radio cell as is possible. If consecutive frames in one radio are similar, then also the interference situation for neighbour radio cells is similar. This means also that the interference situation is less varying. To achieve this goal the following simple steps have to be performed. The scheduler stores the start and the end of its own transmissions in a frame. In the next frame, after the scheduling has been performed, an additional step is inserted. Within the DL phase and also within the UL phase the order of transmission of the scheduled PDU bursts can be chosen without any restrictions. The scheduler tries to find the order of transmission in the DL phase as well as in the UL phase that gives the most resemblance to its previous frame. In the current implementation it does this by brute force. It calculates all possible permutations of the transmission order of the scheduled bursts and calculates a measure for the resemblance between the frames.

The transmission order is chosen which gives the highest resemblance to its previous frame. This ensures that the interference situation varies as little as possible.

In Fig. 4 an example for the unsorted transmission order and for the transmission order with MSI is given. They have been extracted out of one of the simulations performed. On behalf of a clear representation only the downlink phase is shown but the same behaviour applies for the uplink phase.

Please notice that the scheduling is performed as given by any possible algorithm. Only after the scheduling has been performed, does the scheduler try to find the transmission order that gives the maximum similarity between the frames.

3 Scenario and Simulation Environment

In this paper the following scenario is considered. It consists of a big exhibition hall with 16 APs and a site to site distance of 62.5 m. There are 8 frequencies available which means that one frequency is used by two APs. Per radio cell, a number of active Mobile Terminals (MTs) are moving around with a speed of 3 km/h. Each active MT sustains a bi-directional connection with the AP. In a TCP connection, 75% of the generated user load is in downlink direction and 25% is in uplink direction. In a UDP connection, 50% of the generated user load is in downlink direction and in uplink direction each.

The attenuation of signals is calculated via the following one slope model for LOS propagation in indoor large open spaces:

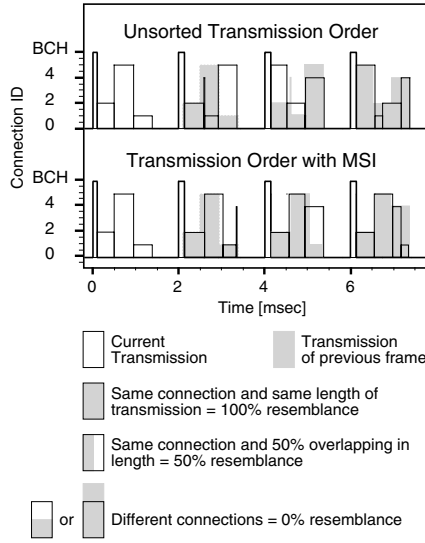


Fig. 4. Comparison between unsorted transmission order and transmission order with MSI

$$L_d[\text{dB}] = 46.7 + 24 \cdot \log(\text{distance}/1\text{m}) \quad (1)$$

Adjacent channel suppression is assumed to be so high that adjacent channel interference can be neglected. Up till now no power control is used and all terminals send with equal transmission power. Applying equation 1 and the distance to the interferer, the C/I value is calculated. Furthermore, log-normal fading with a standard deviation of 7 dB is added in order to model shadowing caused by e.g. people moving around. A model which was proposed in [4] is applied. It uses the following correlation function with a decorrelation length d_{corr} of 3.5 m [5]:

$$R(\Delta x) = e^{-\frac{|\Delta x|}{d_{corr}} \cdot \ln 2} \quad (2)$$

According to files generated out of link level simulations [6] the calculated C/I corresponds to a PER which is then applied to this PDU.

In the current state of the simulation perfect measurement of the C/I values is assumed for the link adaptation. The AP and the MTs store the last N C/I values of every connection. These values constitute the basis for the decision which Phy. Mode is used.

4 Modelling of Co-channel Interference

In order to have a realistic co-channel interference, two radio cells are implemented which interact with each other. In both radio cells detailed implementations of the protocols for AP and MTs are used. The sources generate data

which is transmitted via TCP/IP respectively UDP between the terminals. In the convergence layer the TCP or UDP segments are segmented to fit into User-PDUs (U-PDUs). The U-PDUs are then transmitted via the wireless link. An Selective Repeat ARQ scheme with bitmap acknowledgements is implemented in detail as described in [7] with a limited ARQ window size (128 in the presented simulations). The collision resolution for the RCH is implemented in detail as described in [7]. The scheduling of acknowledgements and data is performed in every frame on PDU basis.

No simplifications are made with respect to the described protocols.

5 Results

With the scenario described above, simulations are carried out to evaluate the approaches described above.

In order to show how much the delay for delay critical connections can be improved with RB and ASP-BLI and placing the delay critical connections in an area of the frame where interference will usually be lower than in other areas of the frame the following scenario was used. It consists of the above described scenario with a mix of UDP and TCP connections. There are 10 active bi-directional connections per radio cell. 12% of the load are generated by the UDP sources and 88% by the TCP sources. In average 55% of the frame are used.

In Fig. 5 and Fig. 6 the delay for the UDP connections is shown for RB with ASP-BLI and ESP. Although the length of the unused capacity, the length of the downlink and uplink phase and the length of the BCH varies from frame to frame it can be seen that ASP-BLI significantly improves the situation for delay critical connections. Down- and uplink delay are significantly better for ASP-BLI than for ESP. The improvement of the delay performance is stronger in the uplink direction than in downlink direction. If PDUs get lost in uplink direction, the MT has to inform the AP first about additional capacity requests before they can be transmitted. This leads to a further increase of the delay if PDUs get lost. Thus it is very important to reduce the number of retransmissions especially in uplink direction if delay critical connections are concerned. Results are only shown for one radio cell but they are very much the same for the other radio cell.

The following scenario is used to show the influence of MSI on system performance. It consists of 5 active bi-directional TCP connections per radio cell. The generated user load refers to pure user data *without any overhead*. Due to TCP timeouts, segment retransmissions do occur and due to duplicate TCP acknowledgements, fast segment retransmissions do occur, which are not counted as generated load. Also, no TCP/IP, convergence layer or HIPERLAN/2 overhead was included in the generated user load.

In Fig. 8 it can be seen that MSI significantly reduces the retransmission load for ASP. In order not to overload Fig. 8 the curves for SSP-MSI and ESP-MSI are not shown. The relative improvement of MSI for SSP is similar to that of ASP. For ESP the improvement with MSI is very small and hardly visible. This is due

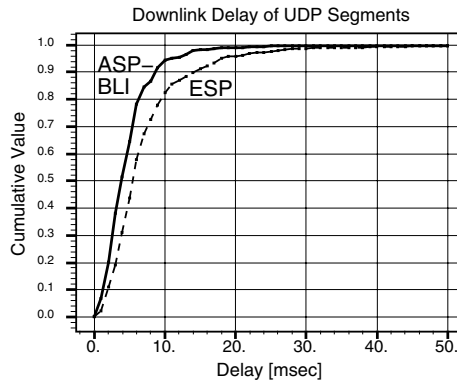


Fig. 5. CDF of downlink delay of UDP segments for RB with ASP-BLI and ESP

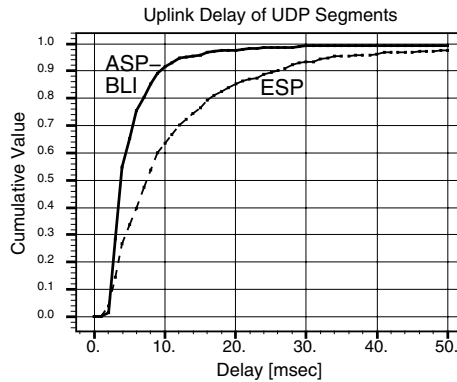


Fig. 6. CDF of uplink delay of UDP segments for RB with ASP-BLI and ESP

to the fact that ESP leads to a high variance in the interference situation. Taking a look at the C/I cumulative distribution function (see Fig. 7), it can be seen that ASP and ESP have almost identical C/I distributions (no difference is visible in the diagram), but the difference in the retransmission load between them is significant. This is due to the fact that with ASP the interference situation is less variable than with ESP and link adaptation can work much more effective. This can be explained by the following. The C/I is counted on a PDU basis. For every transmitted PDU the C/I value is measured in the simulations and counted in the C/I distribution function. With ESP, only short silent periods are inserted before the scheduled bursts. For the co-channel radio cell this means that in a PDU burst, there may be some PDUs that overlap with the silent period in the co-channel radio cell and there are other PDUs of the same PDU burst that overlap with transmissions in the co-channel radio cell. This means that in one PDU burst belonging to one connection there are PDUs which have a high C/I value and others which have a lower C/I value. This is a difficult situation for

the link adaptation to cope with since there is no optimal Phy. Mode for this situation. If a too high Phy. Mode is used, many retransmissions do occur. If a too robust Phy. Mode is used, transmission capacity is wasted. With ASP there exists only one long silent period. In the co-channel radio cell will be some PDU bursts which overlap completely with the silent period and others which do not overlap with the silent period at all. Then the PDUs of a PDU burst have either a high or a low C/I value. While the number of PDUs that overlap with the silent period in the co-channel radio cell is almost the same for ASP and ESP (leading to an almost identical C/I distribution), link adaptation can work much more effectively with the situation produced by ASP. This behaviour of ESP makes it impossible for MSI to gain much improvement since the variance in the interference situation produced by ESP is too dominating.

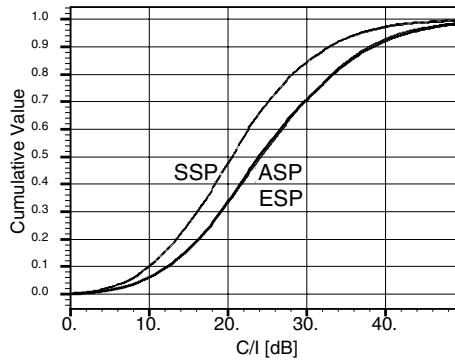


Fig. 7. CDF of C/I for RB with ASP, SSP and ESP

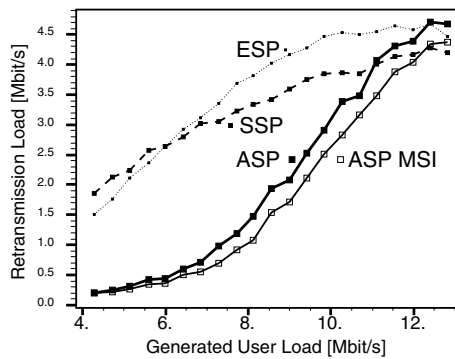


Fig. 8. Retransmission Load for RB and several placements of the silent periods (with and without MSI)

6 Summary and Conclusion

In a HIPERLAN/2 system with co-channel interference, interference and link adaptation are important topics. In order to have a realistic co-channel interference, two radio cells are implemented which interact with each other. In both radio cells detailed implementations of the protocols for AP and MTs are used. Data transmission between the terminals is carried out via TCP/IP and UDP.

It has been shown that co-channel interference can be reduced and link adaptation can become more effective in low and medium load conditions. This is done by limiting the burstiness of transmissions and an asymmetric placement of the silent periods in two co-channel radio cells. With a modified asymmetric placement of the silent periods it is possible to improve the interference situation in certain areas of the frame for *both* up- and downlink and in *both* co-channel radio cells. This improvement can be exploited to significantly reduce the delay of delay critical connections which can be scheduled in areas of the frame where interference will be usually lower than in other areas of the frame. These approaches work well for low and medium load conditions but their effect becomes very small in higher load conditions since they depend on the existence of unused capacity in the frame. Another approach was presented which works under all load conditions. With an easy algorithm and independent of any other conditions it optimizes the similarity between consecutive frames in one radio cell. If consecutive frames in one radio cell are similar so is the interference situation for co-channel radio cells. It has been shown that it is possible to significantly reduce the retransmission load under all load conditions if this approach is applied. In this paper an ideal measurement of the C/I values is assumed for the link adaptation. In the future the effect of the described approaches will be investigated for a realistic link adaptation and actual available measures for the radio link quality.

References

1. ETSI BRAN: HIPERLAN Type 2 Technical Specification TS 101 475 V1.1.1, Physical (PHY) layer. (Apr. 2000)
2. S. Deng: Empirical model of WWW document arrivals at access link. International Communications Conference (ICC) (1996)
3. J. Rapp: HIPERLAN/2 System Throughput and QOS with Interference Improving Strategies. Proceedings of IEEE VTC'01 Spring (Rhode Island) (2001)
4. M. Gudmundson: Correlation Model for Shadow Fading in Mobile Radio System. Electronics Letters **27** (November 1991) 2145–2146
5. M. Fiocco, S. Stavrou, and S.R. Saunders: Measurement and Modelling of Shadowing Cross-correlation at 2 GHz and 5 GHz in Indoor Environments. Antennas and Propagation 2000, Davos, Switzerland (April 9-14, 2000)
6. J. Khun-Jush, P. Schramm, U. Wachsmann, and F. Wenger: Structure and Performance of the HIPERLAN/2 Physical Layer. Proceedings of IEEE VTC'99 Fall (Amsterdam) (1999)
7. ETSI BRAN: HIPERLAN Type 2 Technical Specification TS 101 761-1 V1.1.1, Data Link Control (DLC) Layer, Part 1 - Basic Data Transport Function. (Apr. 2000)

An Intra-media Multimode Wireless Communication Terminal for DSRC Service Networks

Masuo Umemoto

Yokosuka ITS Research Center, Telecommunications Advancement Organization of Japan
Hikarino-oka 3-2-1, Yokosuka, Kanagawa 239-0847, Japan
umemoto@yarp-its.tao.go.jp

Abstract. A future ITS info-communication systems using DSRC is discussed. As the first step of an inter-media multimode terminal in the future, an intra-media multimode wireless communication terminal using software defined modem technology is developed in order to adopt several DSRC service networks. The experimental software defined modem can work at a high data rate of 4 Mbps in the $\pi/4$ DQPSK. It enabled multi-modulation-system ($\pi/4$ DQPSK and GMSK), multi-frame-format, and multi-bit-rate.

1 Introduction

Recently, Intelligent Transport Systems (ITS) has become the focus of global attention. Various info-communication services in the ITS must take an important role in safe and comfortable driving, and therefore they are being researched and developed all over the world [1]. The various info-communication services are provided with cellular telephone systems, Dedicated Short-Range Communications (DSRC), and digital broadcasting, as shown in Figure1. Seamless-connection-communication architecture among these networks for ITS info-communication services was reported [2].

However, new services require new on-board equipment. As a result, cars will be possibly filled up with many communication terminals. Furthermore, it is necessary that the content provided by ITS service is revised or expanded with a gradual improvement on infrastructure of ITS info-communication systems. Similarly, as wireless communication technology related to the vehicular terminal equipment is progressively improved, the communication system in the vehicular terminal equipment needs upgrading from an existing system to an advanced one. Therefore, a multimode wireless communication terminal [3][4] using software radio technology [5][6][7] has been proposed and developed.

DSRC service networks, such as electronic toll collection (ETC), local traffic information services, emergency dispatch services, parking payment system and so on, are peculiar to ITS. Therefore, in this paper, as the first step of an inter-media multimode terminal in the future, which can handle a cellular telephone system, several DSRC services and a digital broadcasting, an intra-media multimode wireless communication terminal for DSRC service networks is examined.

State-of-the-art DSRC service systems for ITS in Japan are briefly described in Section 2. Furthermore, specifications and experimental results of an intra-media multimode wireless communication terminal for DSRC service networks are shown in Section 3.

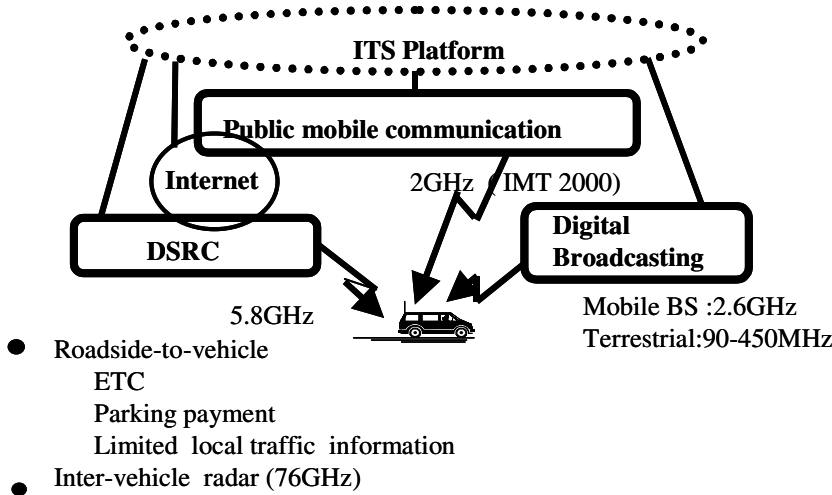


Fig. 1. ITS Info-communication systems in the future

2 DSRC Service Systems

DSRC system is a mobile communication system among ITS info-communications system. The DSRC architecture has been developed according to the ISO-OSI layer model. But due to the real-time constraints a three-layer approach, that is the physical layer, the data link layer, and the application layer, has been chosen. In Japan, a DSRC standard has completed, and it has introduced for the ETC system [8][9]. An outline of the physical layer in the DSRC standard as ARIB (Association of Radio Industries and Businesses) STD T-55 is listed in Table 1. An On-Board Terminal (OBT) uses an active transponder. Through providing different frequencies for up-link and down-link, frequency division duplex is available. In five years, the ETC system will be introduced at about 730 tollgates out of the total 1300 tollgates of Japanese national expressways.

Another DSRC application like road-to-vehicle communications in smart cruise systems [10] has employed an experimental radio specification, in which $\pi/4$ DQPSK modulation system, a data rate of 512 kbps, a information update cycle of 100 msec and an available communication distance of 100 m were used, although 5.8 GHz band and multiple access of TDMA/FDD were used.

VICS (Vehicle Information and Communication System)[11][12] as a traffic information service was operated through radio-wave beacons, infrared light beacons,

and FM multiplex broadcasts. The radio-wave beacons system using 2.5GHz band with a data rate of 64 kbps is a simplex application of DSRC.

Table 1. ARIB STD-T55 for ETC

Item	Specifications
Carrier frequency	5.8GHz band
Transmission power	Roadside station:max 300mW Mobile station: max 10mW
Modulation system	ASK(Amplitude Shift Keying)
Data rate	1024 kbps
Available communication distance	Max. 30 meters
Multiple access	TDMA-FDD system
Access control	Slotted ALOHA protocol

Those are the first generation of DSRC systems. In the ETC system, for example, a TDMA system having 8 (maximum) time slots is adopted for some users in a spot zone, therefore, a data transfer rate per user may become actually around 100 kbps, and communication time is also short because of a spot zone.

To encourage the continued development of DSRC services, techniques for higher data-rate should be studied in the next generation. As a result, the next generation DSRC can accomplish high speed and broadcast communication services. Moreover, the DSRC system will have following features.

1. Effective use of frequencies by using small-zone configurations
2. Large-volume, high-speed information transmissions to moving vehicles
3. Electronic payment through radio communications
4. Services provided by Internet connections

Taking advantage of these features, the widespread development of ITS services based upon DSRC is expected. In North America, they are working on the next generation of DSRC standards at 5.9 GHz band [13]. In EC, a project to combine the extensive functionality of an intelligent in-vehicle terminal with the use of a two-way communication link with the roadside is being developed [14].

Many base stations along roadsides will be connected with DSRC networks, which is one of the most important study-issue in DSRC systems. For example, DSRC network architecture for ITS multicast services [15] was reported.

3 Experimental System of Intra-media Multimode Wireless Communication for DSRC services

The final target of ITS multimode wireless communication terminals is to handle three different kinds of ITS info-communication media, which are cellular telephone systems, DSRC, and digital broadcasting. It will be expressed as an inter-media multimode terminal. However, multiple applications of DSRC will be realized in the near future. An intra-media multimode wireless communication terminal for DSRC service networks should, therefore, be examined as the first step of the final target.

In order to make a feasible check of high-speed information transmissions to moving vehicles, a data rate of 4 Mbps using $\pi/4$ DQPSK and a duplex system were employed in the experimental system of intra-media multimode wireless communication for DSRC services. Furthermore, information transmission using a broadcast type is cost-effective, and it will be used even in the future. Therefore, both the roadside base station A for the high-speed duplex system and the roadside base station B for simplex system were experimentally developed. Both base stations employed different modulation systems and different carrier frequencies. Those experimental systems provided a condition to study multimode and multi-frequency terminals. Figure 2 shows the block diagram of the experimental system containing an experimental multimode wireless terminal as a mobile unit.

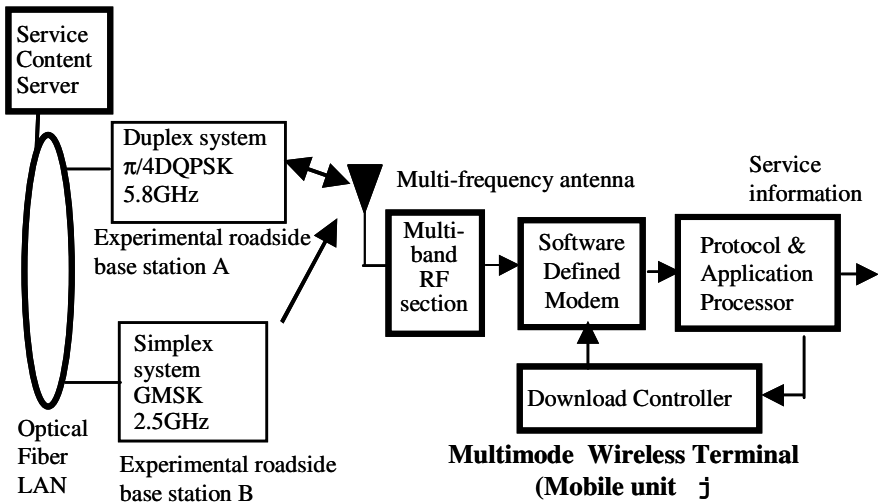


Fig. 2. An experimental DSRC system for the multimode wireless terminal

4 An Experimental Multi-mode Terminal

Basic specifications of an experimental intra-multimode terminal using software defined radio technology in Fig. 2 are listed in Table 2.

Table 2. Basic specifications of an experimental intra-media multimode terminal

Items		Specifications	
Link		Up and down	Down only
Modem	Modulation systems	$\pi/4$ QPSK	GMSK
	Data rate	4000kbps	64kbps
	Signal format	continuous or time division	
	Output signals	I and Q signal	
RF	Carrier frequency	5.8GHz	2.5GHz

A multi-frequency microwave antenna has been developed. The frequencies of both 5.8 GHz and around 2.5 GHz are available. To deal with the multi-frequency signal provided by the antenna, two receiver front-end circuits suitable for multiple-frequency-resonance antenna and common receiver circuit for SNR maximization were employed in the multi-band RF section. The section consists of wide-band amplifiers, multi-band frequency converters, automatic gain controller, and IQ signal interfaces.

Architecture of the OBT for DSRC is constructed with layer 1, 2, and 7, according to the ISO-OSI layer model. In order to realize adaptability in communication systems of DSRC, signal processing related to the communication systems in the layer 1 is necessary to be carried out through software programs. As shown in Table 2, the experimental modem enables multi-modulation-system, multi-frame-format, and multi-symbol-rate. Furthermore, using a download controller in Fig. 1, even a new ITS service in the future will be promisingly available through downloading the modem software of the new ITS service. Thus, the software definable modem system is essential for a seamless-connection-communication architecture [2] in ITS services.

The digital modem is configured mainly with digital signal processors (DSP) and field programmable gate arrays (FPGA). Evolutional development of such semiconductor device will bring an inter-media multimode terminal as the final target.

5 Conclusions

A future ITS info-communication systems using DSRC was discussed. As the first step of an inter-media multimode terminal in the future, an intra-media multimode wireless communication terminal using software defined modem technology was studied in order to adopt several DSRC service networks. The experimental terminal with a high data rate of 4 Mbps were presented. It enabled multi-modulation-system ($\pi/4$ DQPSK and GMSK), multi-frame-format, and multi-bit-rate.

References

1. S. Inada: Future Development of ITS Info-communication Systems. Microwave Workshop digest (MWE'99), pp.15-24, Yokohama, Japan, Dec. 1999.
2. S. Kubota, Y. Ito and T. Munaka: A Study of Info-communication Technologies Required in ITS Networks. Technical Digest of the 1st workshop on ITS Telecommunications, pp.53-58, YRP Yokosuka, Kanagawa, Japan, Oct. 2000
3. H. Harada and M. Fujise: Multimode Software Radio System by Parameter Controlled and Telecommunication Toolbox Embedded Digital Signal Processing Chipset. Conference Proceedings of ACTS mobile communications summit '98, pp.115-120, Greek, June 1998.
4. M. Umemoto et al: Development of a Multimode Wireless Communication Terminals in ITS. Proceeding of 7th World Congress on Intelligent Transport Systems, Paper No. 3104, Turin, Italy, Nov. 2000.
5. J. Mitola: Software Radio. IEEE Communications Magazine, pp.24-25, May 1995.
6. SDR forum: Technical Report version 2.1, Nov. 1999
7. R. Kohno: Perspective of Software Radio: Spatial and Temporal Communication Theory Using Adaptive Array Antenna for Mobile Radio Communications. Microwave Workshop digest (MWE'97), pp.25-31, Yokohama, Japan, Dec. 1997.
8. Electronic Toll Collection System (ARIB STD-T55 Version 1.1 in Japanese), Feb. 1999.
9. Y. Fujimori: Outline of ETC system introduction plan and the standards draft. 6th World Congress on ITS, Toronto, Nov. 1999.
10. www.ahsra.or.jp
11. www.vics.or.jp
12. S. Yamada: The strategy and deployment plan for VICS. IEEE Communications Magazine, pp.94-97, Oct. 1996.
13. L. Armstrong: New DSRC standards under development for north America. 7th world congress on ITS, Paper No. 4022, Turin, Italy, Nov. 2000.
14. P. Perre: Integrating the DSRC link as standard equipment in any vehicle. 7th world congress on ITS, Paper No. 2192, Turin, Italy, Nov. 2000.
15. T. Iwahashi, T. Miyoshi, and J. Inoue: DSRC network architecture for ITS multicast services. 7th world congress on ITS, Paper No. 3244, Turin, Italy, Nov. 2000.

Simulation of Traffic Engineering in IP-Based Networks

T.T. Mai Hoang and W. Zorn

Institute for Operating and Dialog Systems, University of Karlsruhe
Am Fasanengarten 5 - 76128 Karlsruhe, Germany
{hoangmai, zorn}@ira.uka.de

Abstract. A practical function of traffic engineering in IP-based networks is the mapping of traffic onto the network infrastructure to achieve specific performance objectives in respect of traffic-oriented and/or resource-oriented. The simulation of such traffic engineering function is necessary needed while planning and optimizing the network resources. This paper describes an new model for analytical simulation of the traffic mapping in IP-based networks in respect of both real-time and best-effort traffic. We mainly conceptualize the simulation as an behavioral modeling of network dynamics. We then mathematical formulate this behavioral modeling as two sub-optimization tasks corresponding to both real-time and the best-effort traffic delivering. As an example of our algorithm implementations, an genetic algorithm for real-time traffic mapping is presented. The article also gives some numerical results from practical examples.

1 Introduction

A practical function of traffic engineering in IP-based networks is the mapping of traffic onto the network infrastructure to achieve specific performance objectives in respect of traffic-oriented and/or resource-oriented [1]. Traffic-oriented performance objectives relate to improvement of QoS provisioned to internet traffic. Resource-oriented performance objectives relate to the optimization of the network utilization. The simulation of such traffic mapping is necessary needed while planning and optimizing the network resources. Most of the present network simulation models focus in the first line on the analytical simulation of traffic engineering in traditional data networks in which the traffic is only considered as one class of "best-effort" service and it is defined by means of a demand metric describing the traffic requirements between all pairs of switch-to-switch or router-to-router. In such approaches, the traffic is modeled only as a bundle of data flows regardless of priorities and QoS requirements [2, 3, 4].

In comparison with data-oriented applications, the real-time applications own special characteristics, which are still not considered in the present approaches. In particular, these applications are typical less elastic and less tolerant of delay variation than data applications. Such applications require the guarantees of capacity constraints (e.g. peak rates, mean rates, burst sizes) and QoS constraints (e.g. packet loss, delay) from the Internet site. Thus, these constraints of the real-time traffic should be considered in the traffic engineering simulations.

This paper describes an model for analytical simulation of both real-time and best-effort traffic mapping in IP-based networks. The remainder of this paper is organized as follows: In section 2, the model for simulating the traffic mapping is presented. In section 3, an genetics algorithm for real-time traffic mapping is described as an example for our algorithm design. The result of some practical examples is shown in section 4. Finally, the conclusion and future works will be given in section .

2 The Simulation Model

The simulation of traffic mapping is formulated as follows:

Given are

- a fixed IP-based backbone infrastructure, the capacities and buffering of its components, the node locations and the link topology, K classes of real-time traffic by means of traffic parameters and QoS requirements, 1 class of "best-effort" traffic by means of given demand metric and packet delay requirements.

To determine

- an cost-effective utilization of the real-time and best-effort traffic on the given backbone.

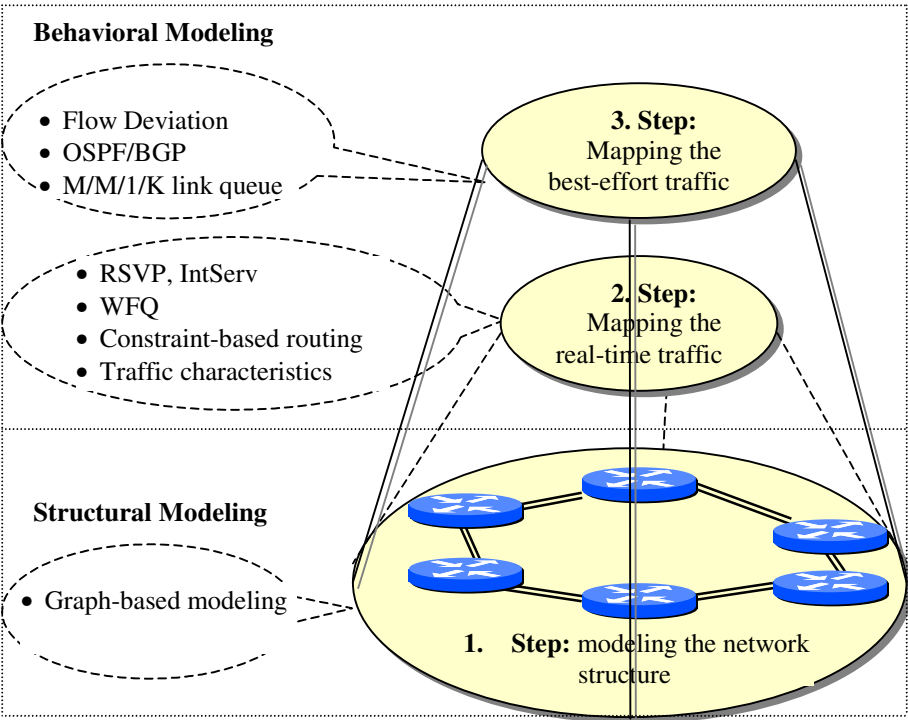


Fig. 1. An overview of the hybrid simulation model for the traffic engineering

This simulation problem is conceptualized, on the one hand, as an structural modeling, and, on the other hand, as an behavioral modeling of the IP network (figure 1). The structural modeling focuses on the abstract representation of IP backbone as a graph $G = (V, E)$ in which V is a set of nodes and E is a set of edges. A node represents a network router and a edge represents a physical link connecting one router to an another one. The behavioral modeling deals with the simulation of the traffic mapping in respect of network dynamics, such as, the traffic characteristics, the RSVP, the MPLS, the constraint-based routing, the OSPF and the resource allocation [5,6,7,8,9]. In this paper, we concern with the simulation of traffic mapping at the IP layer relating to Integrated Services architecture. Whereby, the goal is to find an optimal network utilization for a given traffic and QoS requirements so that the cost of the bandwidth needed will be minimal.

For this traffic mapping, we use the following simple scheduling discipline, namely, the full priority is given to the real-time traffic, and the channel capacities and node capacities that are not used by the real-time traffic are made available to the "best-effort" traffic. Based on this assumption, the traffic mapping consists of two steps. First, the real-time traffic is tried to be mapped onto the given fixed IP backbone. After all real-time traffic demand is mapped, the "best-effort" traffic is then mapped to this backbone under consideration of his spare capacity.

2.1 Mapping the Real-Time Traffic

The mapping of real-time traffic is an analytical simulation of its transmitting over real IP-based networks. An such transmitting process can be seen as an sequence of network mechanisms carried on heterogeneous set of Integrated Services routers, running different routing protocols and using different forwarding algorithms. These mechanisms deals with the Integrated Services Architecture [5,7] and relate to path selection, traffic characteristics, resource allocation, and, traffic assignment to the established path which may have been selected by routing protocol or by some other means.

The path selection is done via Integrated Services routers which perform constraint-based routing, admission control and resource allocation based on the information contained in the source traffic specification and in the desired service specification which is generally named as traffic characteristics. In this paper, the traffic characteristics are used as an input for our traffic mapping algorithms.

Constraint-based routing refers to a class of routing systems that compute routes through a network subject to satisfaction of a set of constraints and requirements [4,5]. Constraints may include bandwidth, hop count, delay and resource class attributes. The other importance mechanisms simulated in our work are the RSVP and the resource allocation. With RSVP, the application source (sender) transmits a path message along the routed path to the unicast or multicast destination (the receiver) [3]. The purpose of the Path message is twofold: to mark the routed path between the sender and the receiver and to collect information about the QoS viability of each router along the path. The resource allocation is an mechanism to determine efficient bandwidths to be reserved on the routers and on the connections along a path computed by constraint-based routing.

The main principle of the real-time traffic mapping is described in figure 2.

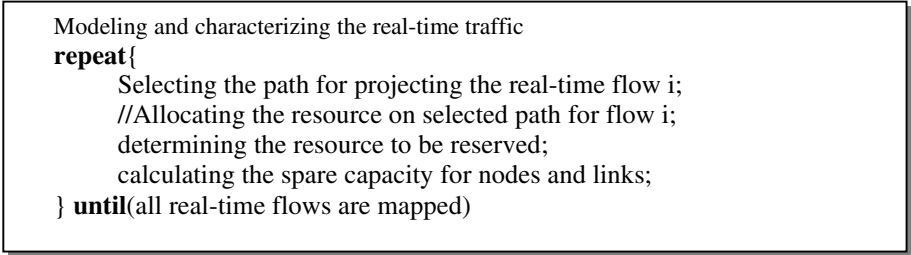


Fig. 2. Principle of real-time traffic mapping

The traffic mapping includes three steps: characterizing the real-time traffic, selecting the path to project the flows, and allocating the resource on the selected path. To characterize the real-time traffic at IP layer, we use the traffic specification T_{spec} and the service specification R_{spec} to model the real-time flows. The traffic specification is mainly based on a simple token bucket (LB), a peak rate p , a minimal policy m , a maximal packet size M . The token bucket has a bucket depth, b , and a bucket rate, r . The service specification defines the rate R to be reserved for a given real-time flow. To schedule packets for outbound transmission from routers, we use the assumption that the Integrated Services routers in our modeled network infrastructures support WFQ which is used today for guarantee QoS in IP-based networks.

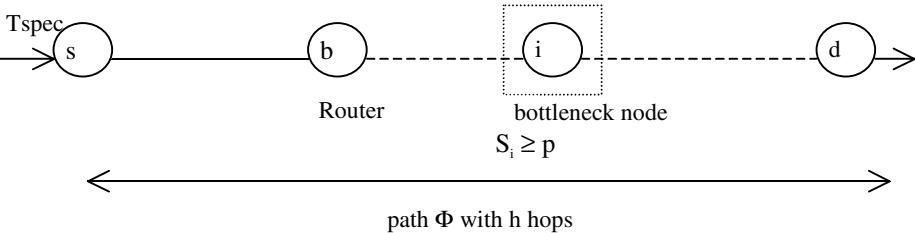


Fig. 3. Selecting the path from s to d for mapping the traffic characterized by T_{spec}

The goal of path selection for a given pair of source and destination node is to find a cost-effective path used for projecting the real-time traffic demands between these nodes on the given fixed IP backbone. Our path selection algorithm is an hybrid simulation of the RSVP and the constraint-based routing mechanism. The algorithm starts at the receiver and searches the next neighbor of the actual router towards the sender (Fig. 3). An subtask developed within this algorithm is an simulation of the admission control which determines whether the router has sufficient available resources to apply the QoS request. The difference to the conventional Depth First Search and OSPF BGP is that our algorithm considers the T_{spec} , the R_{spec} , the spare capacity on the routers and on the links as well as the hop number while selecting the path. Furthermore, in our algorithm, the IP routing for real-time flows are taken into consideration. This algorithm is developed as genetic algorithms.

The resource allocation on the selected path Φ is formulated as follows. Given are the traffic specification T_{spec} , a selected path Φ consisting of p nodes and the parameters described above. To find the resource $R=\{R_1, R_2, \dots, R_j, \dots, R_p\}$ to be allocated at nodes i along Φ under consideration of the spare capacities S_i at nodes i belong Φ , and the upper bound d_0 of end-to-end delay. This problem is then modeled as an local optimization problem as follows:

To optimize

$$F = \sum_{i \in \Phi} \frac{S_i}{S_i - R_i} + F_0 \quad (1)$$

Subject to

$$R_i \leq S_i \text{ and } d_{reqd} \leq d_0 \quad (2)$$

Whereby, d_{reqd} is the desired queue delay bound for the path Φ , F_0 is the constants calculated using the traffic parameters and the spare capacities at the nodes. To find are the resource R_i to be allocated at nodes $i \in \Phi$ so that the cost function F will be minimal. This optimization task is a non linear combinatorial optimization problem that is solved using the lagrangean procedure and the convexity characteristic of the cost function. The rate to be allocated at the nodes i belonging to Φ can be computed at follows: $R_i = \max\{r, R_i\}$.

After the reserved rate R is determined, the spare capacity of each node and of each link belong to path Φ are recalculated. Based in this backbone spare capacities, the best-effort traffic mapping is done using the flow deviation method described in the next section.

2.2 Mapping the Best-Effort Traffic

The "best-effort" traffic mapping is an analytical simulation of the best-effort traffic transmission over packet networks. This transmission can be seen as the approximation of the OSPF Border Gateway Protocol (BGP). Our "best-effort" traffic mapping is based on the assumption that the packet arrival process is a poisson process with exponential distribution and the links are modeled as independence M/M/1/K FCFS queue. The traffic mapping is formulated as the following optimization task:

To optimize

$$T = \frac{1}{\gamma} \sum_{i \in E} \frac{f_i}{c_i - f_i} \quad (3)$$

Subject to

$$T \leq T_0 \text{ and } \frac{f_i}{c_i} \leq 1 \quad (4)$$

Whereby

$$f_i = \frac{1}{\gamma} \sum_{i \in E, i \in p, p \in P} \gamma_p \quad (5)$$

Where, T is the total network delay based on Kleinrock assumption and Little's form[4]. f_i is the flow on link i . P is the set of paths between any pair of demand nodes. p is a path in P . γ_p is the traffic demand on the path p . γ is the sum of all traffic demand entering the network.

We solve this optimization task using the well known method named "flow deviation" (FD) [2]. The FD method is based on the observation that if a small amount of flow is moved from a path with larger incremental delay to a path with smaller incremental delay, then the total delay is decreased. The incremental delay on a path is just the sum of the incremental delay of all links in the path. The FD method proceeds by assigning lengths to the links, based on their incremental delay, then finding the shortest path from each source to each destination, based on these length, and mapping the traffic demand to the links based on these paths. Thus, the flow pattern found is then superposed with the previously found flow patterns. That means, a new flow pattern is formed by sending path of the flow on the old paths and part of the flow on new path. The amount of the flow deviated to the new path is chosen to minimize the total delay. The FD is implemented in our model as an genetic algorithm.

The result of the traffic mapping process is the network throughput which can be used to obtain predictions and formulate control strategies under various conditions as well as to guide network upgrading plans.

3 Genetic Algorithms for Real-Time Traffic Mapping

Genetic algorithm (GA) is optimization technique based on the mechanisms of genetic adaptation in biological systems [10]. The algorithm maintains a population of all possible solutions to the given problem. In one of the most commonly used representations, a solution in the search space is a string represented by a sequence of characters or of numbers. This solution string is called the chromosome. The quality of an chromosome is judged by its fitness values that indicate which chromosomes have a better potential of being carried to the next generation.

A genetic algorithm starts with an initial population of solutions and simulates the process of evolution. After a number of generations, highly fit chromosomes will emerge corresponding to good solutions to the given problems.

Four main steps in development of an genetic algorithm are the Chromosomal coding-schema, Reproduction/selection, Crossover and the computation. The first step

is the process of encoding the chromosomes. The second step concerns with the selecting of potentially good strings from the current generation to be carried to the next generation. The crossover is the process of shuffling two randomly selected strings to generate new offstrings. The last step, the Computation, relates to the calculating of the fitness value using objective function.

The population size is finite in each generation of GA, which implies that only relatively fit chromosomes in generation i are carried to the next generation ($i+1$). The process of selection, crossover, and mutation is repeated till the termination condition is satisfied.

Our traffic engineering simulation presented in the last sections is based on two algorithms running tandem, one for real-time traffic mapping and one for best-effort traffic mapping. In this section, the genetic algorithm GA1 for the real-time traffic mapping will be outlined in detail.

For the implementation of GA1, networks are modeled in the form of chromosomes. This chromosome is a list of real values describing spare capacities of routers and links, and the fitness values. One of the fitness values is the hop number and the other are the minimum of the spare capacities of nodes and the minimum of the space capacity of the link.

Chromosomal coding-schema. In GA1, we use two types of chromosomes, namely, the Network Chromosome (NC) and the Path Chromosome (PC). A NC Chromosome describes the actual capacity reserve of the whole IP-based network. The first n fields of the NC chromosome describe the spare capacities of the corresponding nodes. The next m fields of the NC chromosome describe the space capacities of the corresponding links. The fitness value is described in the last three fields of this NC chromosome. These fields are the node number, the minimum of the node spare capacities, and, the minimum of the link spare capacities. At each time of the computation process, a new NC chromosome can be computed via crossover/mutation between the old NC chromosome with the best PC chromosome. In this case, the new NC chromosome is then replaced to the old NC. A PC Chromosome describes the capacity reserves belonging to a possible path from a given source node s to a given destination node d . This path can be used for delivery the given real-time traffic demand from s to d . In comparison with a NC chromosome, in a PC chromosome, only the fields describing the nodes and links belonging to the given path are set to equal to its spare capacities. The other fields are set to equal to null.

Initialization. At the begin of the GA1, the initial NC chromosome describes the input resource of the IP-based network. The set of PC chromosomes is set initial as null. For each real-time traffic demand from node s to node d , the GA1 first determine a initial set of PC chromosomes. Each of these chromosomes represents the available resource on a possible path from s to d .

Computing the fitness value. For each PC chromosome found above, the fitness is calculated and set into the last three fields of this chromosome.

Reproduction and Selection. In a initial set of PC chromosomes and for a given traffic demand, the chromosomes having the minimal node capacity or minimal link capacity less than sum of the peak rates of real-time flows belonging to this traffic demand are first removed from the set. Only the PC chromosomes having minimal fitness hop number are kept to the set. After that, the chromosome having minimal node capacity is selected to the next generation. We name this chromosome as

PC_best which means that the chromosome for the traffic demand from node s to node d is chosen. For PC_best, the rate R to be reserved on the selected path is determined.

Crossover. For each iteration in GA1, two crossover actions are done. The first crossover is carried on the chromosome PC(s,d) in which the non zero fields of PC(s,d) are overwritten by the value of the rate R . The second crossover is done via subtracting the NC chromosome to the PC(s,d) chromosome.

Termination condition. GA1 is terminated if all traffic demands are removed from the real-time traffic demand set T_r . Namely the set T_r is null. This means that all real-time traffic demands are mapped onto IP-based backbone. The routing for IP flows are considered during the genetic operations. An overview of the algorithm GA1 is shown in figure 4.

```

begin GA1
NC  $\leftarrow$  Initialization;
 $T_r$  is set of real-time traffic demands;
while ( $T_r \neq \text{null}$ ) do
  begin
    select  $t_{s,d} \in T_r$ ; //  $T_r = \{\text{source } s, \text{ destination } d, \text{ characteristics of real-time flows}\}$ 
    PC_set  $\leftarrow$  null;
    begin
      PC_set  $\leftarrow$  Finding a initial set of PC Chromosomes belonging to  $t_{s,d}$ ;
      Computing the fitness value of all chromosomes in PC set;
      //reproduction, selection:
      PC_best  $\leftarrow$  Selecting the best PC chromosomes to be in PC_set;
      Calculating the rate  $R_i$  to be reserved from PC_best;
      for all  $v \in \text{PC\_best}$ : IF  $v \neq \text{zero}$  THEN  $v = R_i$ ; //Crossover 1
      NC  $\leftarrow$  NC - PC_best; //Crossover 2
       $T_r \leftarrow T_r - \{t_{s,d}\}$ 
    end;
  end;
end GA1

```

Fig. 4. Genetic algorithm for real-time traffic mapping

4 Numerical Experiments

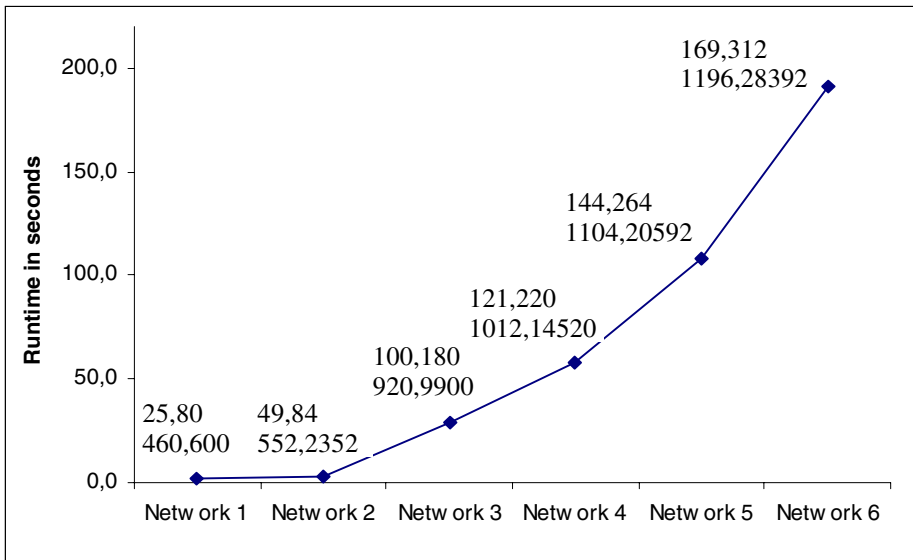
Our model is successfully applied for simulating the traffic engineering in IP-based network infrastructures. Several IP backbones with different sets of routers, links, traffic demands and traffic characteristics are used as the tested IP-based networks.

Table 1. Characteristics of selected real-time flows

Traffic characteristic	64 Kb/s voice	Video Conference	Stored video
bucket depth b [kb]	0.1	10	100
bucket rate r [Mbps]	0.064	0.5	3
peak rate p [Mbps]	0.064	10	10
max. packet size M [kb]	0.1	1.5	1.5
End-to-end delay bound	50	100	100

Table 2. The tested IP-based network structures

Network name	Node number	Link number	Number of real-time traffic	Number of best-effort traffic
Network 1	25	80	460	600
Network 2	49	84	552	2352
Network 3	100	180	920	9900
Network 4	121	220	1012	14520
Network 5	144	264	1104	20529
Network 6	169	312	1196	28392

**Fig. 5.** The runtime of the simulation algorithm

In our experiments, we consider one class of best-effort traffic and three classes of real-time traffic as the input traffic sources. The best-effort traffic is characterized via an traffic metric. Each element of this metrics is a tuple describing the IP traffic demands in packet number and in volume unit (Mbps) to approximate the traffic from one router to one other router. The real-time traffic is shown in table 1. The input data for the tested IP backbone infrastructure is described in table 2.

The simulation model is implemented in Java Jbuilder under Window NT. The computation was carried out on an Intel Pentium Processor 333 and finished within a few minutes. Figure 5 shows the runtime of the simulation algorithm. The runtime depends on the complexity of the IP backbone infrastructure (table 2), the number of real-time and best-effort traffic demands, and the number of genetic strings per generation as well as total number of evolution.

Conclusions and Outlook

The main contribution of this paper includes the following. We described the analytical simulation of traffic engineering applying to the mapping of both "best-effort" and real-time traffic onto an IP-based network. We conceptualized this traffic mapping on the one hand as an abstract representation of the network infrastructure, and, on the other hand, as an behavioral modeling of the network dynamics relating to IntServ, RSVP, CR, OSPF BGP and resource allocation. We modeled this dynamical network behavior with two sub-optimization tasks which are solved using genetic algorithms in Java Jbuilder under Window NT. Computational tests with different IP-based backbone topologies were carried out with insightful result. Our work can be used to obtain predictions and formulate control strategies under various conditions as well as to guide network upgrading plans. Our future work deals with the extension of our model for traffic mapping relating to Multi Protocol Label Switching (MPLS).

References

- [1] D.O. Awduche. MPLS and Traffic Engineering in IP Networks. *IEEE Communication Magazine*, December 1999, P. 42-47.
- [2] A. Kershenbaum. *Telecommunication network design algorithms*. McGraw-Kill Computer Science Series, 1993.
- [3] D. Minoli. *Broadband Network Analysis and Design*. Artech House, Boston-London, 1993
- [4] M. Schwarz. *Computer Communication Network Design and Analysis*. Englewood Cliffs. N.J. Prentice-Hall, 1997.
- [5] S. Shenker. Specification of Guaranteed Quality of Service, RFC 2212, 1997
- [6] S. Shenker. General Characterization Parameters for Integrated Service Network Elements, RFC 2215, 1997
- [7] Braden. Integrated Services in the Internet Architecture: an Overview, RFC 1633, June 1994.
- [8] J. Wroclawski. The Use of RSVP with IETF Integrated Services, RFC 2210, 1997
- [9] E. Crawley et al. A Framework for QoS-based Routing in the Internet. RFC2386, 1998.
- [10] D.E. Goldberg. *Genetic algorithms in Search*. Addison-Wesley, Massachusetts, 1989.

An Algorithm for Available Bandwidth Measurement

Jingsha He, C. Edward Chow[†], Jihui Yang[‡] and Takafumi Chujo

Fujitsu Laboratories of America, Inc.

595 Lawrence Expressway
Sunnyvale, CA 94085, USA

Tel: +1 408-530-4576

Fax: +1 408-530-4515

jhe@fla.fujitsu.com

Abstract. In this paper, we present an algorithm for available bandwidth measurement of a path between two hosts as well as some preliminary simulation results. The measurement algorithm is based on active probing with two techniques we have developed: variable speed probing and zoom-in/zoom-out. Compared with previous work, the algorithm has the advantage of low overhead and fast convergence because it relies on the detection of traffic trends (with variable speed probing) rather than any specific properties of probing samples. The measurement can self-adapt to any bandwidth ranges (with zoom-out) and respond to accuracy requirements (with zoom-in). Therefore, no knowledge about the bottleneck bandwidth of the measured path is required. We are currently experimenting with self-similar traffic over a real network environment to gain more experience and to further validate and improve the measurement techniques.

1 Introduction

The available bandwidth of a path between two network points is one of the important dynamic network characteristics used for optimizing resource utilization in traffic engineering and for admission control in quality of service. Since available bandwidth is a dynamic parameter, it must be determined based on both the capacity of the links and the current traffic that the links carry. Consequently, an effective measurement of available bandwidth can also serve as the means of determining the current traffic. Due to the dynamic nature of the traffic, available bandwidth has to be measured more frequently depending on the fluctuation situation of the traffic. That is, the interval between successive measurements should be a parameter to maintain the timeliness and usefulness of the measurement results. This requires that the measurement overhead be kept as low as possible while the desired measurement accuracy can be achieved.

[†] This author's address is Department of Computer Science, University of Colorado, Colorado Springs, CO 80918, USA.

[‡] This author's address is Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA.

In this paper, we present a new algorithm and the techniques for available bandwidth measurement that we have proposed and implemented for evaluation. We also present some preliminary simulation results to validate the measurement techniques. In the algorithm, we do not assume any knowledge about the capacity of the network or links, nor do we assume any knowledge about the traffic. Rather, we explore some distinctive properties of traffic behavior through active probing and detect the changes in the derivation of the measurement results. This algorithm is also very general in the sense that it responds to accuracy requirements and can automatically adapt itself to measure links of any bandwidth.

This paper is organized as follows. In the next section, we briefly review some previous work related to available bandwidth measurement. In Section 3, we describe the algorithm and the techniques. In Section 4, we present and discuss some preliminary simulation results. Finally, we conclude this paper in Section 5.

2 Related Work

Network measurement has been the subject of a number of studies during the past few years. Bolot [2] analyzed end-to-end packet delay and loss in the Internet and used a phase plot to characterize the phenomenon in a congested network. In the study, he showed that, when multiple packets are transmitted at a low speed, the plot for the round trip delay was distributed randomly and, when at a high speed, the plot formed a distinctive pattern of distribution. The explanation for the different patterns can be attributed to network congestion when the packets are transmitted at the high speed. NEPRI [1] is a network measurement tool developed by Fujitsu Laboratories Limited based on this theory in which one or more rounds of probing packets are used and the phase plots are drawn to detect the pattern that corresponds to network congestion. The probing packets in each round are sent at a fixed speed. Based on the phase plot for the current round, the probing speed for the next round is adjusted upward or downward. As soon as a desired pattern is detected and the result is computed satisfactorily, the measurement is finished with the bandwidth result.

Jacobson [5] developed a measurement tool called pathchar for the measurement of bottleneck bandwidth of the links along a path. During the measurement, a number of probing rounds with different packet sizes are used to probe each intermediate node along the path until the desired destination is reached. The tool has been further experimented by Downey [4] and the experience showed that pathchar could yield a reasonably accurate measurement results. However, because pathchar is for deriving the physical characteristics, i.e., the capacity, of the links in the path, the overhead is very high. This is prohibitive for available bandwidth measurement, which has to be performed frequently, not to mention that the method may not be readily applicable to available bandwidth measurement.

Bprobe/Cprobe [3] uses multiple rounds of probing messages to calculate the bottleneck and the available bandwidth of a path. The measurement result is computed based on all the probing rounds at different speeds and with different packet sizes. Therefore, the overhead of the measurement is very high because a fixed number of multiple rounds of probing are always needed. While this may be acceptable for the bottleneck bandwidth measurement, it is required that overhead for available bandwidth measurement be kept at the minimal. In addition, probing at the

bottleneck speed for available bandwidth measurement is a too high price to pay, not to mention the dependency of the measurement on the knowledge about the bottleneck bandwidth.

3 The Measurement Algorithm

We use active probing in the measurement and regard the network that connects the measurement node, or the agent, and the measured node, or the server, as a black box. The agent invokes the measurement and derives the result by probing the server with a series of probing packets, which triggers the acknowledgment packets. The agent also uses a timer for each packet to record its round trip time RTT. We describe how the available bandwidth can be derived from RTT along with some other information about the probing packets in the algorithm.

Let T_s be the sending time of a probing packet and T_r the receiving time of the acknowledgement packet, then the RTT of the probing packet is

$$RRT = T_r - T_s.$$

If there is no congestion, the RTT is about the same for every probing packet. If there is, the RRT may be larger for the probing packet because of additional delay caused by the congestion. Therefore, by comparing the RTTs for different probing packets, we are able to determine if network congestion occurs. On the other hand, since we don't know the minimum RTT, we need to use more than one packet to get different RTTs. The key is to how systematically generate the probing packets that can yield the different and meaningful RTTs. If the RTTs increase, we can conclude that congestion has occurred. However, we still cannot conclude that congestion does not affect the probing packet with the smaller RTT because of the lack of knowledge on the minimum RTT. Nevertheless, the network phenomenon that we can observe in the case of congestion is that of larger RTTs. In general, the worse the congestion gets, the larger the RTT becomes. Therefore, the only practical way to measure the available bandwidth is to cause a brief congestion to a link in the path. This requires that sufficient probing traffic be generated and sent through the link to cause the congestion.

We now describe the basic techniques in the algorithm for available bandwidth measurement: variable speed probing and zoom-in/zoom-out.

3.1 Variable Speed Probing

We use a set of packets at a variable probing speed, from low to high, for the agent to probe the server. The optimal range of the speed is one with the property that some later packets would cause link congestion while some early ones wouldn't. In this case, we will be able to observe different RTTs and derive the available bandwidth by detecting the congestion point based on the RTTs. This congestion point gives us the needed information to derive the available bandwidth. This is because, before this point, all probing packets should have the same RTT and, after this point, although the RTTs may be different, the general trend is consistent and the RTTs should be larger than that without congestion. Therefore, to achieve the objective, each measurement round would consist of probing packets that impose different bandwidth

requirements on the link. That is, probing packets of different speeds (with the same packet size) or different sizes (with the same speed) should be used in the measurement. In the illustration, we will use the variable probing speed method, while the variable packet size is equally applicable.

The bandwidth requirement on the link by a packet can be determined by its size S and the time interval t between it and the next:
$$S/t.$$

If we let the time interval between packets P_i and P_{i+1} is t_i , $1 \leq i \leq n-1$, and $t_i > t_{i+1}$, the bandwidth requirement of packet P_i is
$$S/t_i.$$

Therefore, given the set of n probing packets $\{P_1, P_2, \dots, P_n\}$ of size S but decreasing time intervals, the bandwidth requirement of the packets increases as they are emitted into the path.

After collecting the RTTs at the probing agent, we use curve matching between the one for the sending probing packets (the sending curve) and the one for the receiving acknowledgement packets (the receiving curve). The sending curve is plotted using the emission time against the packet number and the time of the first packet is set to 0. The receiving curve is also plotted in the same way using the receipt time against the packet number and the time of the first packet is aligned to 0. These two curves are then plotted on the same plain, which is called curve matching.

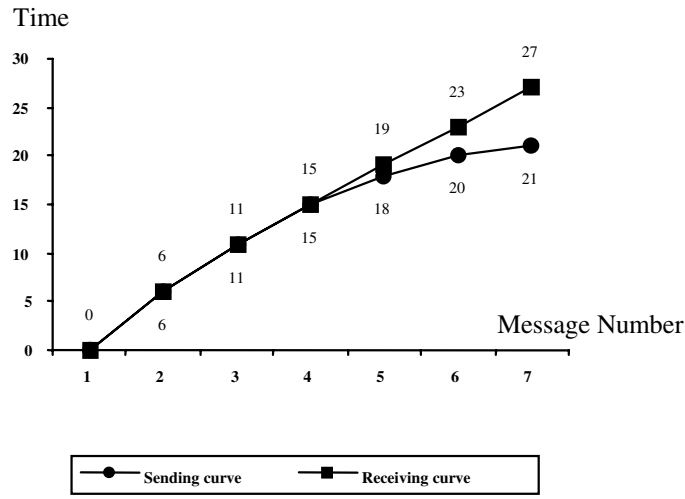


Fig. 1. Available Bandwidth Measurement

We illustrate this measurement technique with an example. In Fig. 1, assume that the 7 probing packets are sent out in the interval starting with 6 and decreasing by 1 at a time. Assume also that packet 4 reaches the speed where the bandwidth requirement causes the link to congest. Due to the congestion, all the subsequent packets will be affected by the congestion and have a RTT equal to or greater than that of packet 4. The RTTs after the congestion point will increase so that the interval between the acknowledgement packets will lag behind that between the corresponding probing

packets. The point in the receiving curve where it starts to diverge from the sending curve is the congestion point. The bandwidth requirement at this point is then used to calculate the available bandwidth.

Traffic fluctuation in the network may cause noise in the RTTs. As the result, the receiving curve may not be as smooth as that in Fig. 1. In such a circumstance, a trend line can be drawn for the receiving curve and estimation techniques can be used by taking into consideration of a number of points around the congestion point. In addition, because the probing packets are discrete, the congestion speed may happen between two adjacent packets. Therefore, using a single point in the calculation of the available bandwidth may cause underestimation or overestimation relative to the actual bandwidth. Consequently, it may be more appropriate to use more than one point in the calculation of the measurement results.

Following is one way to determine the intervals between the probing packets. For simplicity, we assume that the intervals decrease linearly by an equal number. Let the probing packet size be S , the number of packets be n , and the bandwidth range for the probing be represented by a range (B_L, B_H) where B_L is the lower end and B_H the higher end. We further assume that $B_H > 0$, $B_L > 0$ and $B_H > B_L$; otherwise, the bandwidth range is meaningless. From these parameters, we can determine that the time interval decrement is $S(B_H - B_L)/(B_H B_L(n-2))$ and the time intervals can be computed using the formula:

$$t_i = \frac{S}{n-2} \left(\frac{n-i-1}{B_L} + \frac{i-1}{B_H} \right) \quad 1 \leq i \leq n-1.$$

We can also make other assumptions on the relationships between the time intervals and use the same technique to determine the intervals, which we are currently studying as a research topic on probing patterns that could yield better performance in terms of measurement accuracy and probing overhead.

3.2 Zoom-In/Zoom-Out

The zoom-in technique is used when the measurement detects the congestion point but the result doesn't meet the required accuracy. This would happen when the bandwidth range is large and, therefore, the result is computed by using points with large bandwidth differences. To improve the accuracy of the result, zoom-in would invoke a new round of measurement during which the bandwidth range is shrunk to a range around the congestion point. The selection of the new bandwidth range will determine the new B_L and B_H and the values of the time interval t_i , $1 \leq i \leq n-1$. The selection of the new bandwidth range may depend on the quality of the previous measurement. If the previous measurement does not give a good curve to clearly identify the congestion point, we should not shrink the bandwidth range too much due to the risk of missing it altogether in the next round of measurement. Because of this limitation, multiple rounds of measurement invoked by zoom-in may be needed. This situation could happen if the previous bandwidth range is too large, the number of probing packets is too small, the accuracy requirement is too high, the network traffic is too volatile, among other factors. Therefore, we use an automatic procedure in the algorithm to determine whether zoom-in is needed after every round of probing and

measurement. Each zoom-in should bring the measurement closer to the final goal and the number of rounds eventually invoked is determined by the number of probing packets n , the size of the packet S , the accuracy requirement, the initial bandwidth range (B_L , B_H) and the fluctuation situation of the traffic. A noisy network could result in more rounds before the desired result can be obtained. The zoom-in process can be viewed as a necessary procedure to improve the measurement result to meet the specified accuracy requirement.

The zoom-out technique is just the reverse procedure of zoom-in. The purpose of zoom-out is to locate the congestion point when the measurement does not identify one. Since we don't have any knowledge about the bottleneck bandwidth, we don't always know what the highest bandwidth should be. Therefore, this mechanism can dynamically expand the bandwidth range to make the measurement algorithm adaptable to any network bandwidth. Zoom-out basically enlarges the measurement area by extending the bandwidth range. It could also move the bandwidth range downward if the previous measurement probes too fast or upward if too slow. The enlargement and adjustment of the bandwidth range could be used together as well. In the algorithm, zoom-out is invoked when the measurement does not identify a congestion point. Totally matching curves indicate that the probing speed is too low while totally diverging curves indicate that the probing speed is too high. Similar to zoom-in, a single zoom-out may still fall short. Therefore, we use an automatic procedure to determine if zoom-out is needed through the examination of the measurement result. The zoom-out can be viewed as a necessary procedure to locate the congestion point used in the calculation of the available bandwidth.

3.3 The Measurement Algorithm

The algorithm combines the variable speed probing and zoom-in/zoom-out to make it a flexible measurement algorithm. The algorithm can be summarized as follows.

- (1) Using n probing packets of size S and picking a bandwidth range based on past measurement or any knowledge or guess about the bottleneck bandwidth, invoke the basic probing technique and curve matching to detect the congestion point. The selection of the bandwidth range does not affect the fundamentals of the algorithm but only the overhead because a bad selection could result in more rounds of probing.
- (2) If the congestion point is detected through curve matching, calculate the result and determine its accuracy. If the specified or system default accuracy requirement is met, the measurement is finished. The bandwidth requirement of the probing packet at the congestion point is the result for the available bandwidth. The available bandwidth could also be computed by using more than one probing packets around the congestion point to neutralize the impact of traffic volatility during measurement.
- (3) If the congestion point is detected but the accuracy requirement is not met, the zoom-in procedure is invoked to determine a smaller bandwidth range for the next round of probing and measurement. The algorithm then continues by looping back to (1).
- (4) If the congestion point is not detected, the zoom-out procedure is invoked to determine a larger or a different bandwidth range through examination of the current measurement. The algorithm then continues by looping back to (1).

From the above discussion, it is clear that the extra overhead of the measurement beyond the basic probing results from the high requirement on the quality of the measurement and from the flexibility of automatically adapting the algorithm to measure any bandwidth, both of which are desirable features in any measurement algorithms. Without the accuracy requirement, the zoom-in procedure could be avoided and, without the need for automatically adapting to any bandwidth, the zoom-out procedure could be avoided. No previous work can achieve the same functionality and flexibility as that of the zoom-in and zoom-out, not to mention the additional overhead. In terms of completeness and performance, we believe that our measurement algorithm is superior to all the previous work.

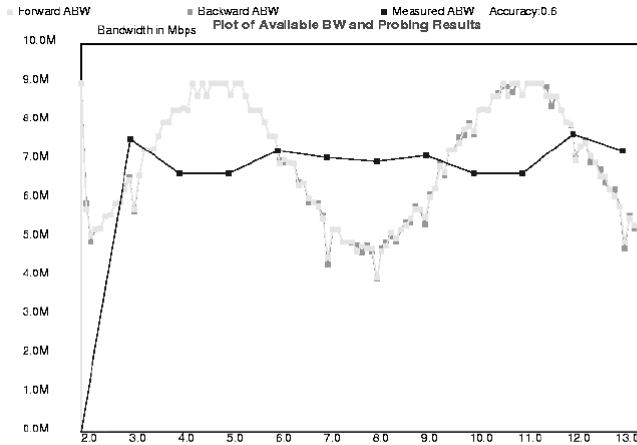


Fig. 2. Simulation Result with 60% Accuracy

4 Implementation and Simulation

We have implemented the presented algorithm and are currently evaluating its performance through simulation and experimentation. We intend to fine-tune the algorithm through this effort and hope to gain more valuable experiences and draw some conclusions regarding available bandwidth measurement in general and the measurement algorithm and techniques presented in this paper in particular. At this moment, we are able to report some limited preliminary simulation results and will expect to have more complete and comprehensive results in the near future along with the lessons learned from the effort.

The simulation results presented here are generated based on the following parameters:

- (1) The number of probing packets, i.e., n , are 30.
- (2) The size of the probing packets, i.e., S , is 1024 bits.
- (3) The initial bandwidth range, i.e., (B_L, B_H) , is (500Kbps, 10Mbps).

- (4) The intervals for the probing packets are determined using the formula for an equal decrement in time.
- (5) The actual available bandwidth fluctuates but always below 10Mbps.

We attach here two simulation results with the accuracy requirements of 60% and 70% in Fig. 2 and Fig. 3, respectively. In both cases, the measurement is done multiple times and a curve is drawn based on the measurement results (in darker color) together with that for the actual available bandwidth (in light color) for the purpose of comparison. We can see that the measurement results for the 70% accuracy requirement are much better than those for the 60% accuracy. However, the overhead is higher, which, on the average, incurs 3.2 rounds of probing for the 70% accuracy vs. 2.5 for the 60% accuracy.

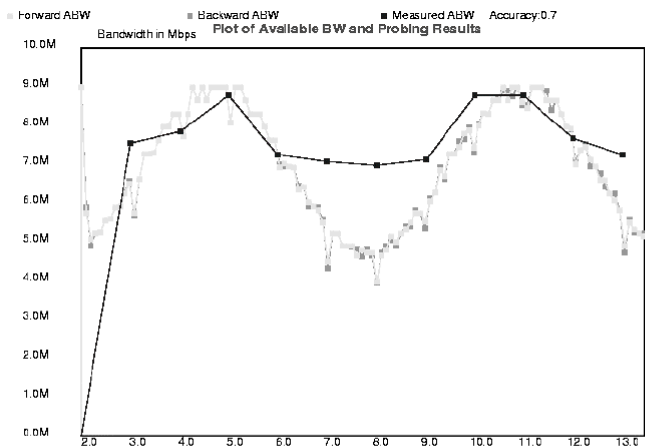


Fig. 3. Simulation Result with 70% Accuracy

We are currently doing more simulation and will fine-tune the algorithm based on the experience learnt from the simulation. In addition, we are incorporating the self-similar traffic pattern into the simulation system and will soon start the simulation using this more realistic traffic pattern for the Internet. The present and future enhancement to the measurement algorithm and techniques as well as to the simulation environment will enable us to learn a great and get more insight understanding of the Internet traffic dynamics and measurement capabilities.

5 Conclusion

We presented a new algorithm for the measurement of end-to-end available bandwidth. The measurement uses the active probing approach with a number of packets emitted at an increasing speed. By systemically collect and compare the probing packets, we can detect whether network congestion occurs, which is the basis for the derivation of the available bandwidth. We described the measurement

techniques and discussed the various issues that make the algorithm adaptable to any network bandwidth and different measurement requirements. We extended the basic measurement technique by introducing the techniques of zoom-in and zoom-out to achieve the objectives. We also presented some preliminary simulation results to show that the algorithm performs as expected. We are currently doing extensive simulation and will move on to live experimentation to further fine-tune the algorithm and improve the measurement performance. We expect to report more complete and comprehensive simulation and experimentation results in the near future along with the lessons and experiences we will have learnt throughout this research and development effort.

References

1. Adachi, Motomitsu, Shinji Kikuchi and Tsuneo Katsuyama, "NEPRI: Available Bandwidth Measurement in IP Networks," *Proc. 2000 IEEE International Conference on Communications*, New Orleans, LA, June 2000.
2. Bolot, Jean-Chrysostome, "Characterizing End-to-End Packet Delay and Loss in the Internet," *Journal of High Speed Networks*, vol. 2, No. 3, December 1993, pp. 305-323.
3. Carter, Robert L. and Mark E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *Technical Report BU-CS-96-006*, Boston University, 1996.
4. Downey, Allen B., "Using pathchar to Estimate Internet Link Characteristics," in *Proc. ACM SIGCOMM'99*, 1999.
5. Jacobson, Van, pathchar, <ftp://ftp.ee.lbl.gov/pathchar/>, April 1997.

Influence of Network Topology on Protocol Simulation

Damien Magoni and Jean-Jacques Pansiot

Université Louis Pasteur / LSIIT
Pôle API, Boulevard Sébastien Brant, 67400 Illkirch, France
{magoni, pansiot}@dpt-info.u-strasbg.fr

Abstract. Simulation is one of the most widely used techniques for designing network protocols. A simulation framework provides a sandbox where a harmful design flaw can easily be detected and removed. This is done prior to implementation and experimentation in an operational environment as it is easier and cheaper to carry out. However, simulation results can be distorted if the simulation model is unrealistic. In particular the topology model used by a protocol simulation can have a great impact on the results. In this paper we present a comparison of the results of an oriented multicast protocol simulation performed on some of the major topology models currently in use in the network research community.

1 Introduction

The aim of this paper is to highlight the impact of network topology on network protocol simulation. The wide use of simulators such as *ns* [14] or GloMoSim [12] by the scientific community for designing network protocols enforces the need of realistic modeling at all levels. The topology models used in simulators have been quite simple since the beginning of network simulation but today's computing power makes simulation possible over larger topologies (i.e. graphs). That's why the use of small graphs following grid-like or random models should be changed in favor of bigger and more realistic graphs.

Section 2 gives an outline of previous studies on the influence of network topology on protocol simulation as well as an overview of the existing topology models. Section 3 presents some properties of the Internet topology and exhibits the characteristics of the topology generators that we will use. Section 4 briefly describes the oriented multicast routing protocol that we will evaluate by simulation. Section 5 shows the influence of the topology models on the protocol simulation results for a typical use of our oriented multicast protocol.

2 Related Work

The influence of topology on protocol simulation results was already noticed in 1993 by Doar et al. The efficiency of their multicasting algorithms was reduced

by 50% when using random graphs rather than hierarchical graphs [4]. In 1994, Wei et al. found that the average node degree of the topology model had an influence that could bias results by 30% when comparing the traffic concentration in core-based multicast trees and in shortest path multicast trees [16]. Later in 1997, Zegura et al. showed that the average delay ratio of center to shortest path was increased by a factor that could go up to 100% when using transit-stub graphs rather than random graphs [17]. Recently Radoslavov et al. did a thorough study of the impact of topology on protocol design [13]. They studied three well known network topology models (i.e. Waxman, Tiers and ITM) and their influence on four multicast protocol paradigms (i.e. multicast trees, forwarding state aggregation, endsystem multicast and alternate path routing). They reported significant result differences depending on the topology model used. Also in a recent study Palmer et al. evaluated their STORM multicast algorithm with 4 topology models, including PLOD and Waxman [11]. Although the average packet overhead was roughly the same (for a 50-client or above topology) whatever the generator used, the plot of the distribution of the percent of protocol overhead per node was highly tied to the type of generator. This paper is an extension of these previous studies. We examine an oriented multicast routing protocol which is very dependent on the underlying topology and has not been already studied in this context. We also use, in addition to the others, a new topology generator that matches more closely the properties of the Internet and that has not been tested in the previous studies (i.e. BRITE).

Concerning network topology models, a well-known early model was defined by Waxman in 1988. This model places the nodes randomly on a plane and then creates links between nodes with a probability depending on the nodes' euclidean distance [15]. The Waxman model belongs to what we call flat topology class. Circa 1996, two new generators were created, namely Tiers [3] and ITM [?]. Both are based on a structured network creation process designed to match the Internet architecture. The ITM topology model is called transit-stub because it is based on the Autonomous Systems' structure. It has been widely used in network simulation tools (e.g. it is distributed in ns). Tiers is based on the LAN-MAN-WAN structure of the Internet. Tiers and ITM generators are both belonging to what we call the hierarchical topology class. Recently new generators were created to build graphs that follow the power-law properties of the Internet. Some of them are already available for testing, namely BRITE [10] and Inet2 [6]. These generators belong to the power-law topology class. In our paper we will only deal with the generators of the last two classes.

3 Network Topology

A network is typically modeled as an undirected graph. The topology of the graph is a description of the way the nodes are connected together. Properties, such as the average node degree and the diameter, give information on a graph topology. For network protocols, the knowledge of the topology of the medium

is very important as it directly translates into useful information such as path length and path redundancy.

3.1 Internet Topology

An accurate knowledge of the topology of real networks is necessary to design graph generators. Recently the topology of Internet has been investigated a lot and new results have been discovered. In particular, some topological properties were found to comply with power-laws. For example, Faloutsos et al. discovered that the node degree distribution of the Internet topology complies with two power-laws [5] (both at the AS-level and at the router-level). The exponents of these power- laws concisely describe their corresponding distributions. The trees' part of Internet has also been studied by Magoni et al. who discovered three power- laws that apply to the size and depth distributions of the trees [8] (at both AS and router levels).

3.2 Topology Models

Table 1 shows the most common topology models currently in use by the research community. As software packages, they are all freely available except PLOD.

Table 1. Topology models

Class	Model(s)	Date & references
Flat topology	Waxman	1988 - [15]
Hierarchical topology	Tiers	1996 - [3]
	Transit-stub	1996 - [17]
Power-law topology	BRITE	1999 - [10]
	Inet2	2000 - [6]
	PLOD	2000 - [11]

In the flat topology models, the edges are created with a probability depending on the distance of the corresponding nodes. In the hierarchical topology models, subgraphs modeling network parts are first generated by using a flat topology method (as in Transit-Stub) or a spanning tree method (as in Tiers). Then the subgraphs are connected together in a way that enforces a multi-level tree-like structure. In the power- law topology models, the edges are distributed to the nodes in a way that matches the skewed node degree distribution of the Internet. This can be done by reverse engineering (as in Inet2 and PLOD) or by the use of preferential connectivity and incremental growth (as in BRITE).

We run the simulation only on the graphs of the last two classes. The flat topology class and the Waxman model in particular has already been widely studied and its drawbacks are well-known. Furthermore, we don't test Inet2 because this generator has been designed to create AS-level topology graphs.

4 Protocol Simulation

We want to evaluate the influence of the topology model used on the simulation results of our agent search protocol. As it is not fully defined yet, we will call it an algorithm rather than a protocol through the rest of this paper. This algorithm has been described in a previous study [9]. In short, our algorithm is an improvement over the expanding rings search mechanism. We want to find agents that are located between a given source and destination. We assume that the initiator of the search knows the address of the destination. Packets are multicasted in a controlled way, so that they do not go too far off the shortest path from the initiator to the destination. Each packet contains a range field that indicates how many hops the packet is allowed to do when it is out of the source-destination shortest path.

Our agent search algorithm is based on an oriented multicast algorithm that is very sensitive to the underlying network topology. This oriented multicast algorithm has also been described in a previous study [7]. We compare our agent search algorithm to the expanding rings search (ERS) algorithm. The expanding rings search has been described in protocols such as YAM [2] and QoSMIC [1].

5 Influence of Topology

In this section we give the results of the simulation of the agent search and ERS algorithms for each of the topology model tested. We also explain how we got the results (i.e. how we set the parameters of the generators and the simulator).

5.1 Simulation Parameters

Table 2 gives the parameter settings of the generators. 20 graphs by topology generator have been generated. Each graph has 2000 nodes and contains 1% of agent nodes. Each algorithm (ours and ERS) has been tested on 500 different source-destination pairs, for four given source-destination distances, for each graph. So for a given source-destination distance, each algorithm has been tested 10000 times.

The simulations have been carried out with the network manipulator software. It is a static network simulator that we have implemented in our laboratory. It is static because it does not take into account any temporal aspect of the communications. The results of these simulations have been merged to give average results. We made these simulations for source-destination distances of 4, 8, 12 and 16 hops. For the ERS, the TTL is increased by 2 while no optimal agent is found, starting at 1 up to a maximum value of 7 (i.e. 1, 3, 5, 7). For our algorithm, the range is increased by 1 from 1 to 4. It is possible for the algorithms not to find any optimal agent because they have to stop their search at some point.

Table 2. Parameter Settings for the Generators

Generator	Parameters	Value(s)
Transit-Stub	#transits in top	20
	#nodes/transit	4
	#stubs/transit	2
	#nodes/stub	12
	Edge method	Waxman
	Alpha	0.5
	Beta	0.5
Tiers	NW	1
	NM	220
	NL	0
	SW	680
	SM	6
	SL	0
	RW	6
BRITE	HS	1000
	LS	100
	m	1, 2
	Node placement	Random
	PC	Only
	IG	Active

5.2 Simulation Results

In this section, we present the simulation results of four variables of interest. These variables are the bandwidth usage, the number of optimal agent found, the number of attempts to find an optimal agent and the efficiency. For each variable, we calculate its value by using our agent search algorithm and by using the ERS algorithm. We divide the former value by the latter to obtain a ratio that enables an easier comparison. Furthermore, as we carried out tests on four different distances, we calculate the average of the four ratio values. So we have one ratio value left for each of the generators used.

Figure 1 shows the bandwidth ratio given by each of the topology models. For example, the Transit-Stub topology model has a value of 1.4. This means that our agent search algorithm creates on average 40% more packets than the ERS algorithm. We can clearly see that there are big differences between the results and that they depend on the type of graphs used for the simulation (i.e. the kind of topology model used). We can already say that the topology model used has a big influence on the results. The biggest gap is a -68% difference that can be found between the Tiers ratio and the BRITE 2 (i.e. with $m = 2$) ratio.

Figure 2 shows the number of optimal agent hit ratio. For example, the Transit-Stub has a value of 1.5. This means that our algorithm finds 50% more optimal agents than the ERS algorithm. The variations between the topology

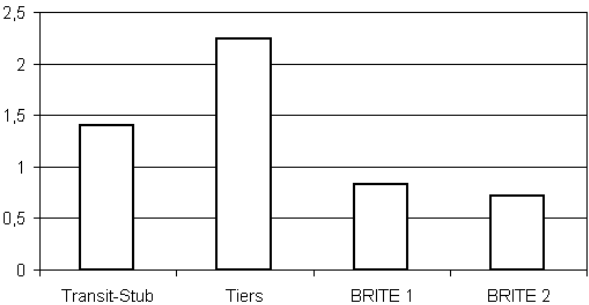


Fig. 1. Bandwidth Ratio

ratio values are of lesser importance than in the previous figure but they are still significant.

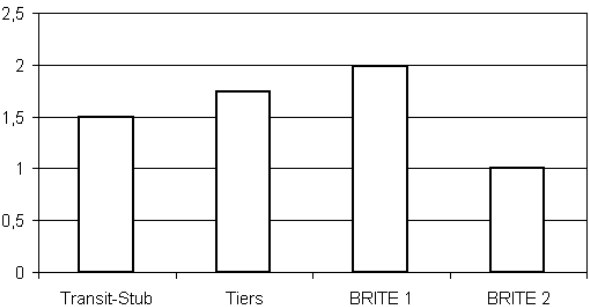


Fig. 2. Optimal Agent Hit Ratio

Figure 3 shows the average number of attempts needed to find at least one optimal agent. The Transit-Stub value of 0.6 means that our algorithm needs on average 40% less attempts to find at least one optimal agent than the ERS algorithm. Here too, the values depend on the topology model.

We have defined a ratio called efficiency to be able to assess the algorithms' performances. The efficiency is equal to the number of optimal agents found divided by the number of packets emitted in the network. As usual we divide the efficiency of our algorithm by the efficiency of the ERS algorithm to obtain an efficiency ratio. Figure 4 shows the efficiency ratio of each topology model. They are all greatly different. Tiers favors the ERS over our algorithm, while the others favor our algorithm. The difference between the Tiers ratio and the

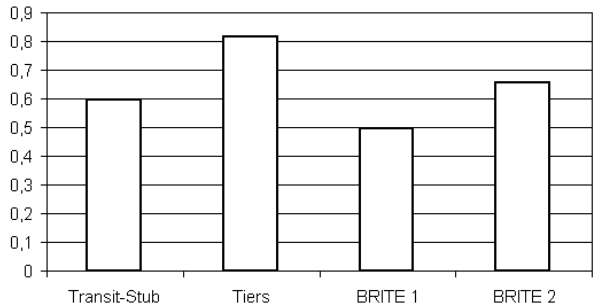


Fig. 3. Attempt Number Ratio

BRITE 1 ratio reaches +198%. The performance of our algorithm over the ERS algorithm is heavily influenced by the topology model used.

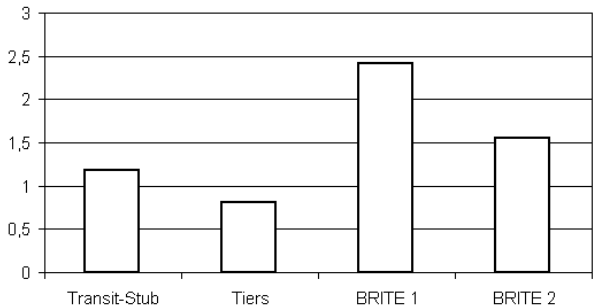


Fig. 4. Efficiency Ratio

6 Conclusion

We showed on a particular multicast algorithm example that the kind of topology model used for protocol simulation has a crucial impact on the simulation results. A protocol performance could be favored by a topology model or disfavored by another. This situation can lead researchers to avoid using simulation for design protocol. Perhaps the best way to draw acceptable conclusions would be to use a topology model that is closest to the real network topology where the new protocol will be deployed. For IP protocols, and routing protocols in particular, the use of the most recent topology models (i.e. of the power-law topology class)

should be recommended. However, there is still room for improvement in creating a topology model that would match the Internet topology. Indeed, even the most up-to-date generators do not take into account all of the Internet topology properties that have been newly discovered. Simulation is such an important tool in network research that it can not be neglected because of a lack of realistic topology generators. It is clear that new enhanced topology models will appear in the near future to reduce the bias owing to topology on protocol simulation results.

References

1. Anindo Banerjee, Michalis Faloutsos, and Rajesh Pankaj. Designing qosmic : A quality of service sensitive multicast internet protocol. In *Proceedings of ACM SIGCOMM'98*, Vancouver, BC, Canada, September 1998.
2. Ken Carlberg and Jon Crowcroft. Building shared trees using a one-to-many joining mechanism. *ACM Computer Communication Review*, 271:5–11, January 1997.
3. Matthew Doar. A better model for generating test networks. In *Proceedings of IEEE GLOBECOM'96*, November 1996.
4. Matthew Doar and Ian Leslie. How bad is nave multicast routing. In *Proceedings of IEEE Infocom'93*, pages 82–89, 1993.
5. Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of ACM SIGCOMM'99*, Cambridge, Massachusetts, USA, September 1999.
6. Cheng Jin, Qian Chen, and Sugih Jamin. Inet: Internet topology generator. Technical Report CSE-TR-433-00, University of Michigan, 2000.
7. Damien Magoni and Jean-Jacques Pansiot. Algorithm for an oriented multicast routing protocol. In *Proceedings of IEEE International Conference on Communications ICC'01*, Helsinki, Finland, June 2001.
8. Damien Magoni and Jean-Jacques Pansiot. Autonomous system level analysis of internet. Technical Report ULP/LSIIT-RR-2001/01, Université Louis Pasteur, 2001.
9. Damien Magoni, Jean-Jacques Pansiot, David Paté, and Dominique Grad. Agent search by oriented multicast. In *Proceedings of ACIS International Conference on Software Engineering Applied to Networking & Parallel/Distributed Computing SNPD'00*, pages 181–188, Reims, France, May 2000.
10. Alberto Medina, Ibrahim Matta, and John Byers. On the origin of power laws in internet topologies. *ACM Computer Communication Review*, 2000.
11. Christopher Palmer and Gregory Steffan. Generating network topologies that obey power laws. In *Proceedings of IEEE GLOBECOM'00*, San Francisco, California, USA, November 2000.
12. Parallel Computing Laboratory, <http://pcl.cs.ucla.edu/projects/glomosim/>. *GloMoSim*.
13. P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, and D. Estrin. On characterizing network topologies and analyzing their impact on protocol design. Technical report, University of Southern California, 2000.
14. The VINT project, <http://www.isi.edu/nsnam/vint/>. *network simulator ns-2*.
15. Bernard Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 69:1617–1622, December 1988.

16. L. Wei and D. Estrin. A comparison of multicast trees and algorithms. In *Proceedings of IEEE Infocom'94, Toronto, Canada*, June 1994.
17. Ellen Zegura, Kenneth Calvert, and Michael Donahoo. A quantitative comparison of graph-based models for internetworks. *IEEE / ACM Transactions on Networking*, 5(6):770–783, December 1997.

An Adaptive Flow-Level Load Control Scheme for Multipath Forwarding^{*}

Youngseok Lee and Yanghee Choi

School of Computer Science and Engineering, Seoul National University
{yslee, yhchoi}@mmlab.snu.ac.kr

Abstract. Compared with the traditional single path routing model, multipath routing increases total network utilization and end-to-end performance. When disseminating traffic into multiple paths, routers should adaptively allocate flows to each path in order to achieve load balancing among multiple paths, as most IP flows are short-lived and the flow size is not normally distributed. Moreover, routers should distribute packet streams belonging to a flow into the same next-hop not to cause end-to-end performance degradation. This paper proposes an adaptive multipath load control method using a flow classifier which detects long-lived flows through the flow characteristics of the duration and the size. By dividing flows into long-lived and short-lived, congestion from the bursty transient flows may be avoided. It is shown by simulation experiments with the real packet trace that the proposed algorithm adaptively controls the load of multiple paths satisfying the given load ratio, and the minimal per-flow states at routers can be maintained by aggregating flows with the destination network prefix.

Keywords: Flow, load control, multipath

1 Introduction

A router capable of multipath routing maintains multiple next-hop nodes for the same destination in its routing table. Multipath routing provides increased bandwidth and enhances the utilization of network resources more than the traditional Internet routing mechanism based on the single shortest path algorithm.

Multipath routing has been incorporated in several routing protocols. The best-known one is the Equal-Cost Multi-Path(ECMP) routing. This is explicitly supported by Open Shortest Path First(OSPF) and Intermediate System to Intermediate System(IS-IS). Some router implementations allow equal-cost multipath for Routing Information Protocol(RIP). In the Multi-Protocol Label Switching(MPLS) network, where IP datagrams are switched by looking up the fixed-size label, paths between an ingress router and an egress router are explicitly set up by Explicitly Routed Label Distribution Protocol(ER-LDP) or

^{*} This work was supported in part by the Brain Korea 21 project of Ministry of Education, in part by the National Research Laboratory project of Ministry of Science and Technology, 2001, Korea.

the Resource ReSerVation Protocol(RSVP). Therefore, multiple explicit Label Switched Paths(LSPs) between an ingress router and an egress router can be set up and there can be even non-shortest paths for multipath routing.

When forwarding packets to multiple paths, routers should have an adaptive load control function for load balancing across parallel paths in order to support dynamic traffic behaviors and varying link/path characteristics(available bandwidth, delay, and packet loss rate). Otherwise, some of multiple paths may experience significant congestion due to the high traffic load.

In this paper, we propose a simple flow classifier based algorithm as the flow-aware adaptive multipath load control scheme. The proposed algorithm has the following features.

- Flow-level load control of multiple paths when the load ratio for each path is given: The input traffic can be split to satisfy the pre-defined load ratio of each path in a flow-level multipath forwarding mode. The sequence of IP packet streams should be maintained within a flow. Otherwise, the receiver must handle out-of-order packet arrivals with a large buffer, and end-to-end performance will be degraded.
- Minimal per-flow states: The number of per-flow states retained by a router should be as small as possible.
- Differentiation between long-lived flows and short-lived ones: [3] suggests that long-lived flows have less bursty arrival characteristics than short-lived flows. The bursty transient flows can abruptly increase the queue length at routers, causing packet losses.

The organization of this paper is as follows. In Section 2, the related work for the multipath and traffic engineering is explained. Section 3 presents the flow-level adaptive load control problem in multipath forwarding. Then, Section 4 describes the proposed flow-level load control algorithm. The results of the performance evaluation are discussed in Section 5, and the conclusion and future work are given in Section 6.

2 Related Work

There have been many studies on multipath routing. [4] proposes a multipath forwarding extension scheme for the distance vector and the link state routing protocol. In [7], Quality-of-Service(QoS) routing via multiple paths for the time constraint is proposed when the bandwidth can be reserved, assuming all the re-ordered packets are recovered by the optimal buffer at the receiver, which causes the overhead of the dynamic buffer adjustment at the receiver. In connection-oriented networks, [8] has analyzed the performance of multipath routing algorithms and has shown that the connection establishment time for multipath reservation is significantly lowered. [9] has proposed a dynamic multipath routing algorithm in connection-oriented networks, where the shortest path is used under light traffic conditions and multiple paths are utilized as the shortest path becomes congested.

To avoid the negative effects by the bursty short-lived flows, the enhanced routing scheme separating long-lived and short-lived flows is proposed in [10] where long-lived flows are dynamically routed whereas transient flows are forwarded on the pre-provisioned paths. However, the flow trigger is considered only under the static network provisioning policy. In [12], a hashing-based load control method without flow states is proposed, but the load adaptation scheme for the dynamic network and traffic behavior is not well presented. In [11], it is shown that the quality of services can be enhanced by dividing the transport-level flows into UDP and TCP flows. Yet, it does not consider the aggregated flows.

3 The Flow-Level Load Control Problem

In this section, we examine how packet-level multipath forwarding may degrade the end-to-end throughput and why the adaptive flow-level load control method should be devised for multipath forwarding.

3.1 Negative Impacts on End-to-End Performance by Packet-Level Multipath Forwarding

Packet-level multipath forwarding in a round-robin fashion may cause the end-to-end performance degradation.

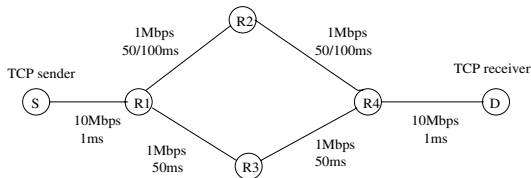


Fig. 1. The simulation topology

When the router $R1$ distributes incoming packets destined for D to two next-hops ($R2$ and $R3$) concurrently (Fig. 1), the effect of the different delays on TCP performance is illustrated in Fig. 2. Fig. 2-(a) represents the case where both the upper path ($R1 - R2 - R4$) and the lower path ($R1 - R3 - R4$) are set to 100 ms, and S sends packets to D after opening an FTP connection¹. Fig. 2-(b) is for the same FTP connection run under different delays (the upper path set to 200 ms). In Fig. 2-(b), the congestion window ($cwnd$) at S periodically decreases by half due to fast retransmit and fast recovery algorithms, resulting in the poor TCP throughput. When two paths have different delays, packets with higher sequence number may arrive at the receiver too early, causing the

¹ This simulation was tested for TCP Reno with NS-2[13]

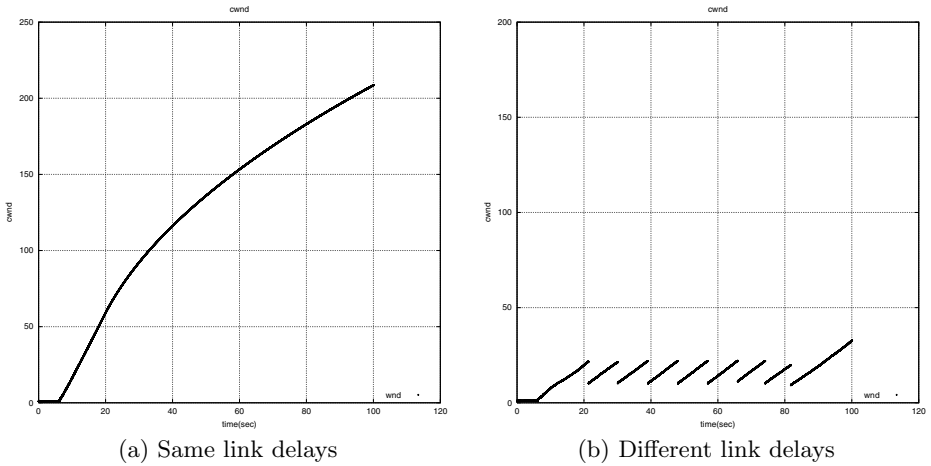


Fig. 2. TCP congestion window behavior under different path delays in packet-level multipath forwarding

receiver to send duplicate ACKs. After receiving three duplicate ACKs, the sender retransmits the late arrived packet again and reduces *cwnd*. In addition to the three duplicate ACK problem, the increasing speed of *cwnd* is slow because the ACKs with lower sequence numbers, which may arrive at the sender later than the ACKs with higher sequence numbers, are ignored.

3.2 Skewed Flow Characteristics

Most IP flows² are shown to be short-lived and small, whereas a few ones have long duration and large traffic loads, dominating the total traffic load in a link or path[2]. Hence, we examine the load balancing condition in general flow-level multipath forwarding.

Assuming that packet arrivals are modeled as packet trains[5], multiple paths are identical, and the packet size is normally distributed, then the flow-level round-robin load balancing can be explained by the following lemma, which is defined for load balancing by multiple identical servers in [6].

Lemma 1. (*Flow-level Round-Robin Load Balancing*): Let l_i be a random variable describing the total delivery time required for all the flows mapped to a given path P_i . Let r' be a random variable of the delivery time for a flow, N be the number of packets, and N' be the number of flows in the batch or train. If N' flows are assigned to m paths in a round-robin manner, then the square of the

² IP flows are defined by packet arrivals satisfying the end point specification(network addresses, transport protocol, and application port) within a time interval.

coefficient of variation of l_i is given by

$$CV[l_i]^2 = \left(\frac{m}{N'}\right) CV[r']^2 \quad (1)$$

and hence, when r' has finite variance and $N' \approx N$

$$\lim_{N \rightarrow \infty} CV[l_i] = 0. \quad (2)$$

From the above lemma it is concluded that for sufficiently large packet train size, the loads in a multiple path set are balanced if the coefficient of the variation of this normal distribution tends to zero.

N' and r' are dependent on the flow organization in a batch or train. The number of flows N' for the given N packets varies from 1 to N . Therefore, when a few flows carry most of the packets ($N' \ll N$), the load balancing can not be achieved, because N' is quite small compared to the large N especially when the flow granularity is coarse.

When f_i and r_i denote the number of packets and the delivery time of a packet for a flow i respectively, the flow delivery time r'_i will be $f_i \cdot r_i$. The expectation of the flow delivery time r'_i is as follows:

$$E[r'] = E[f] \cdot E[r] \quad (3)$$

Therefore, the square of the coefficient of variation of the flow delivery time will be as follows.

$$CV[r']^2 = \frac{Var[r']}{E[r']^2} = \frac{Var[f] + Var[r']}{E[f]^2 \cdot E[r']^2} \quad (4)$$

Thus, $Var[f]$ should be finite in order for the flow delivery time r' to have a finite coefficient of variation. However, the skewed flow size distribution may result in a very large variation. In Fig. 3³, for example, even 1 % of flows contain 65 - 90 % of the load in byte percentage, and 57 - 88 % in packet percentage.

4 The Proposed Load Control Scheme

We develop control scheme for routers with two next-hops(a primary path and a secondary one) for the same destination. This scheme can be easily extended to multiple next-hop cases.

4.1 Flow Classification

For flow assignment, flows which have long duration, high-bit rate, and large flow size(called "base" flows) are distinguished from short-lived transient ones, and assigned to the primary path. Fig. 4 depicts the ingress router with the flow classifier. Packets not belonging to base flows(called "transient" flows) are forwarded to the secondary path.

³ This trace was measured for one hour on KORNET, a commercial Korean Internet backbone, by Cisco NetFlow[14].

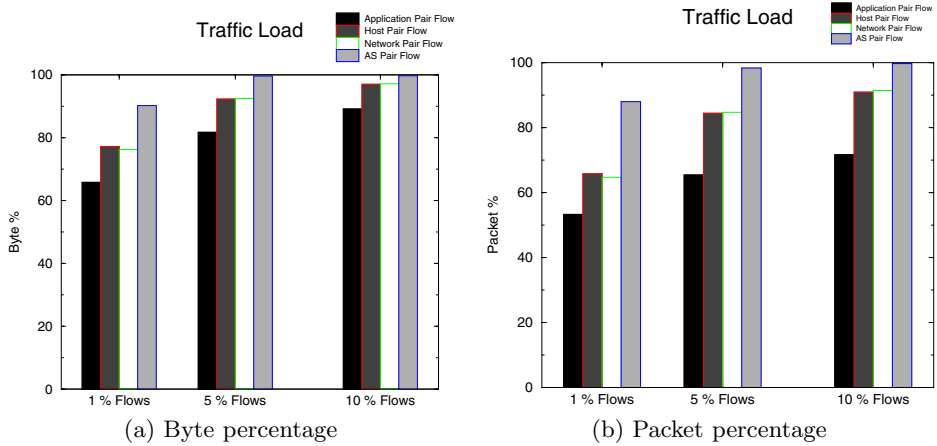


Fig. 3. Traffic load distribution of 1/5/10 % flows

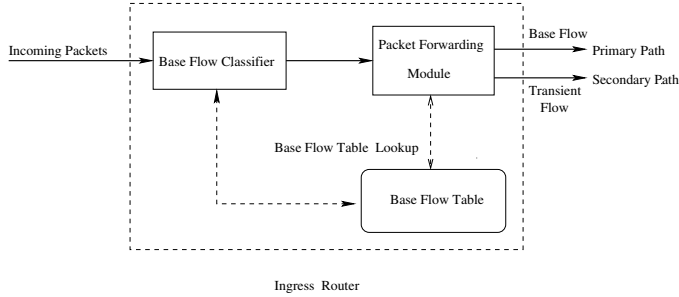


Fig. 4. The ingress router with load control

The base flow detection is based on the X/Y(X: packet count, Y: timeout) flow classifier used in IP switch[1]. In the X/Y flow classifier, a flow is detected when X packets with the same flow specification arrive within Y seconds. This means that the initial X packets of a base flow are forwarded to the transient path. By adjusting X and Y , we can easily control the load assigned to each path. If we increase X (or decrease Y), then less flows will be detected and the load to the primary path will decrease. Decreasing X (or increasing Y) will do the opposite. Thus, adaptive X/Y flow classifier can adapt to dynamic path and traffic behaviors. The packet forwarding module delivers an incoming packet to an appropriate next-hop by looking up the flow table.

4.2 Load Control Algorithm

The load ratio of the primary path is measured by the number of packets sent along the primary path over the total number of packets. The load control algorithm uses the adaptive base flow classifier to meet the given load ratio of the primary path. Although there are two possible adaptive parameters in the X/Y base flow classifier, the flow size, X , is chosen to be variable. The flow size X of the adaptive base flow classifier is adjusted according to the most recent base flow load($BFL(t)$). If $BFL(t)$, which is smoothed by the previous value($BFL(t-1)$) and the recent sample($SampleBFL$), is greater than the given base flow load threshold(BFL_{thr}), the flow size X is increased by Δ . Otherwise, the flow size is decreased multiplicatively by the pre-defined constant, C . Δ is set such that the base flow size estimator X does not increase too quickly. A constant k is used to adjust the increasing amount of Δ .

$$\Delta = \frac{X}{k}, (k > 1) \quad (5)$$

The most recent base flow load in the interval $[t-1, t]$ uses the first-order filter to dampen the abrupt fluctuation of the base flow load. When α approaches 1 ($0 < \alpha < 1$), abrupt changes are suppressed.

Algorithm 1. Adaptive Load Control Algorithm

```

1:  $BFL(t) = \alpha \cdot BFL(t-1) + (1 - \alpha) \cdot SampleBFL$ 
2: if ( $BFL(t) \geq BFL_{thr}$ ) then
3:    $\Delta = \frac{X}{k}$ 
4:    $X = X + \Delta$ 
5: else
6:    $X = \frac{X}{C}$ 
7: end if
```

5 Performance Evaluation

To evaluate the load control algorithm, packet traces at the border router of our campus network were captured with tcpdump, and the full routing table of the border router was used. The traffic through the border router shows an average of 4 - 5 Mbps and the traditional traffic pattern of TCP(FTP, WWW) applications.

To compare the pre-defined load ratio of the primary path and the detected base flow load, we define the normalized base flow load ratio variation, $\hat{B}(\%)$,

$$\hat{B} = 100 * \frac{|BFL_a - BFL_{thr}|}{BFL_{thr}} \quad (6)$$

where BFL_{thr} and BFL_a are the threshold and the acquired base flow load ratio for the primary path, respectively.

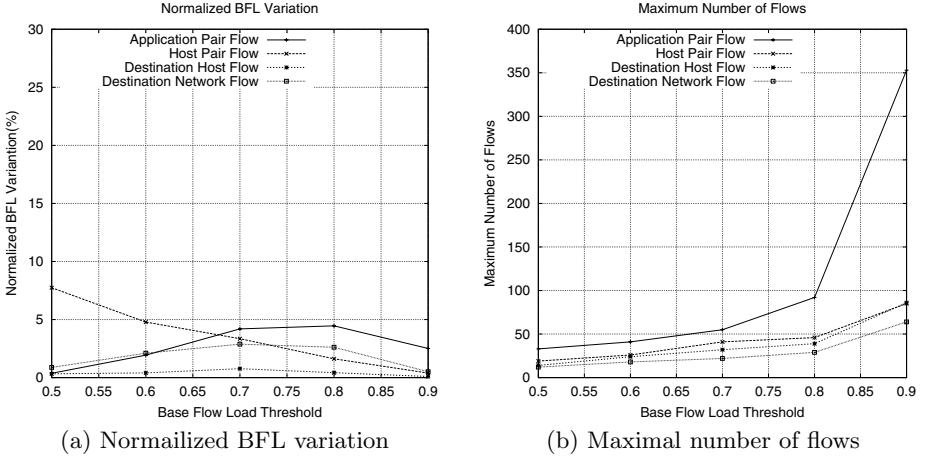


Fig. 5. Normalized base flow load ratio variation and maximal number of flows

The proposed algorithm requires pre-flow state at routers, and not scalable. This is overcome by aggregating flows going to the same destination. The aggregation can be done in different levels: application, host pair, destination host, or destination network.

From Fig. 5-(a) we can see that the proposed algorithm satisfies the base flow load threshold within 10 %. Among four flow aggregation types, the destination host flow mode shows the lowest normalized variation(1 %) under various base flow load thresholds. This is because the variation of the flow size and the flow duration is rather high except the destination host flow aggregation which generates normally distributed flows.

For the proposed algorithm, the ingress router should maintain the entire per-flow states. The maximum number of flows will affect the scalability of the proposed algorithm. In Fig. 5-(b), the destination network prefix aggregated flows require the minimum number of per flow states even at high threshold. In conclusion, we can see that base flow load assigned to the primary path does not deviate much from the given threshold with the minimal memory requirement.

6 Conclusion

In this paper, we proposed an adaptive flow-level load control algorithm for practical multipath forwarding. It is shown by experiment that the proposed algorithm, which uses the adaptive X/Y flow classifier, divides the input traffic

in a way to satisfy the pre-defined load ratio of multiple paths in order to absorb the dynamic flow characteristics. The number of per-flow states required for a multipath packet forwarding router can be minimized by aggregating flows with the destination host or network prefix. Through this load control scheme, the network resource can be fully utilized and the congestion from the bursty transient flows can be avoided. The proposed load control scheme will be useful for multipath packet forwarding without much additional overhead at routers.

References

1. P. Newman, T. Lyon, and G. Minshall, "Flow Labeled IP: A Connectionless Approach to ATM," IEEE INFOCOM'96, 1996
2. W. Fang, and L. Peterson, "Inter-AS Traffic Patterns and Their Implications," Princeton University TR-598-99, March 1999
3. A. Feldmann, J. Rexford, and R. Caceres, "Efficient Policies for Carrying Web Traffic over Flow-switched Networks," IEEE/ACM Transactions on Networking, pp. 673-685, Dec. 1998
4. J. Chen, P. Druschel, and D. Subramanian, "An Efficient Multipath Forwarding Method," INFOCOM'98, 1998
5. R. Jain, and S. A. Routhier, "Packet Trains - Measurements and a New Model for Computer Network Traffic," IEEE JSAC, vol. 4, no. 9, Sept. 1986
6. D. Thaler, and C.V. Ravishankar, "Using Name-Based Mappings to Increase Hit Rates," IEEE/ACM Transactions on Networking, February 1998.
7. N. S. V. Rao, and S. G. Batsell, "QoS Routing Via Multiple Paths Using Bandwidth Reservation," INFOCOM'98, 1998
8. I. Cidon, R. Rom, and Y. Shavitt, "Analysis of Multi-Path Routing," IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 885 - 896, Dec. 1999
9. S. Bahk, M. Zarki, "Dynamic Multi-path Routing and How it Compares with other Dynamic Routing Algorithms for High Speed Wide Area Networks," ACM Computer Communications Review, vol. 22, no. 4, pp. 54-64, Oct. 1992
10. A. Shaikh, J. Rexford, and K. G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," SIGCOMM'99, 1999
11. P. Bhaniramka, W. Sun, R. Jain, "Quality of Service using Traffic Engineering over MPLS: An Analysis," Globecom'99, 1999
12. Z. Cao, Z. Wang, and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," INFOCOM'2000, 2000
13. The Network Simulator- NS-2, <http://www.isi.edu/nsnam/ns/>
14. Cisco, <http://www.cisco.com>

Performance Evaluation of CANIT Algorithm in Presence of Congestion Losses

Hafssa Benaboud and Noufissa Mikou

Université de Bourgogne, Laboratoire LIRSIA, Equipe Réseaux.
BP 47870 - 21078 Dijon Cedex, FRANCE
{benaboud,mikou}@crid.u-bourgogne.fr

Abstract. In this paper, we analyze by queuing-simulation CANIT (Congestion Avoidance with Normalized Interval of Time) algorithm performances in presence of congestion losses. In a former work [3], we proposed the algorithm (CANIT) for TCP (Transmission Control Protocol) congestion avoidance phase in order to improve fairness during this phase, and we showed that using CANIT algorithm in an environment without loss, instead of standard congestion avoidance algorithm improves both congestion avoidance fairness and bandwidth utilization for long RTT connections. In this paper, we consider congestion losses and show that the fairness as well as the bandwidth utilization are more efficient when using CANIT algorithm than the standard one. Moreover, the losses in CANIT algorithm are equivalent to those in standard congestion avoidance.

1 Introduction

TCP (Transmission Control Protocol) is a sliding window protocol which allows the sender to transmit a given number of segments before receiving an acknowledgment (ACK). TCP uses a set of congestion control algorithms: *slow start*, *congestion avoidance*, *fast retransmit* and *fast recovery* [7,11], to control the sliding window size used by the sender and to retransmit lost packets.

The slow start algorithm is used at the beginning of a TCP connection or after a congestion detected by a timeout¹. Congestion Window size (CW_{nd}) is initialized by 1 segment (in practice, CW_{nd} is measured in bytes, usually 512 bytes for one segment, but, to simplify discussion, it is expressed here in terms of segments). For each acknowledgment received, slow start algorithm increases CW_{nd} by one segment, providing an exponential increase of the sliding window size. Slow start phase continues until either CW_{nd} reaches a given *Slow Start Threshold* ($SSThresh$) value or a segment loss is detected.

Slow Start phase is followed by congestion avoidance phase, during which the value of CW_{nd} is greater than or equal to $SSThresh$, and TCP sender

¹ TCP sender detects losses (congestion) in two different ways: by a timeout or by reception of three duplicate ACKs.

increments its current CW_{nd} by $\frac{1}{CW_{nd}}$ each time an ACK is received (here, CW_{nd} is measured in number of segments). Congestion avoidance phase continues until a segment loss is detected. If congestion is detected by a timeout, the congestion window is set to one segment after a retransmission timeout, and the sender proceeds with the slow start. Otherwise, fast retransmit and fast recovery algorithms are used. These last allow TCP to detect and recover from segment drops. For details, see [1].

While these algorithms are very important, they can also have a negative impact on long delay link performance of TCP, for exemple, satellites links [8].

In this paper, we focus on congestion avoidance algorithm. The policy used in that algorithm is unfair when multiple connections with different RTTs share the same resource in the network. In fact, the senders of long RTTs connections make more delay to receive an ACK than those of short RTTs connections. So, these last increase their sliding windows more frequently. Notice that Round Trip Time consists on a segment transfer time and its ACK transfer time. TCP estimates its RTT at the beginning of connection by sending one segment at connection establishment and waiting for its ACK. For details, see [2].

In order to improve the TCP fairness, several solutions are suggested [3,5,4]. In [3], we proposed a new algorithm for congestion avoidance phase, named CANIT algorithm (Congestion Avoidance with Normalized Interval of Time). That algorithm increases, for all connections sharing same network ressources, the size of congestion windows, by approximatively the same number of segments during the longest RTT of these connections. For that, it uses a new parameter NIT (Normalized Interval of Time), which represents an interval of time, during which, each connection must increase its CW_{nd} by one segment. In [3], we consider an environment without losses and show that CANIT is more fair than the standard algorithm. In this paper, we consider a lossy environment and we compare CANIT algorithm performances to those of standard one using one of the configurations used in [3] with finite buffers.

The paper is organized as follows. Section 2 briefly outlines congestion avoidance and CANIT algorithms. Section 3 present our configuration and its queuing model. In section 4, simulation results are used to compare the standard algorithm and CANIT performances in presence of congestion losses . Finally, section 6 presents our conclusions.

2 Congestion Avoidance and CANIT Algorithms

When CW_{nd} becomes greater than $SSThresh$ in slow start phase or after fast retransmit and fast recovery phases, congestion avoidance phase is used [11]. In this phase, TCP sender increments its current CW_{nd} by $\frac{SegSize * SegSize}{CW_{nd}}$

each time an ACK is received where CW_{nd} is measured here in bytes, and $SegSize$ is the size of segments (usually equal to 512 bytes). That means that an CW_{nd} is roughly increased by $SegSize$ per RTT.

As mentioned above, the policy which is used in congestion avoidance phase is unfair when multiple connections with different RTTs share same resources in the network. In fact, waiting for ACK in long RTT connections (for exemple satellites links) takes more time than it makes for short RTT ones. That connections increase then their sliding windows more frequently.

CANIT is proposed in [3] in order to improve congestion avoidance fairness. It increases congestion window sizes for all connections sharing the same network ressources, by approximatively the same number of segments during the longest RTT of these connections. For that, it uses a new parameter NIT (Normalized Interval of Time), which represents an interval of time, during which, each connection must increase its CW_{nd} by one segment. Thus, the following policy is used by all connections:

- For a connection, after reception of an acknowledgment (ACK), TCP sender increments its CW_{nd} by

$$\frac{RTT}{NIT} \times \left(\frac{SegSize \times SegSize}{CW_{nd}} \right).$$

where, RTT is the round trip time of the connection, $SegSize$ is the segment size and NIT is the normalized interval of time.

As mentioned above, in order to make discussion easier, we express CW_{nd} in number of segments. Then, the additive increase can be expressed by

$$\frac{RTT}{NIT} \times \left(\frac{1}{CW_{nd}} \right).$$

Consequently, in absence of losses, TCP sender of a given connection k must receive, at time t , $CW_{nd}(t - RTT_k)$ acknowledgments during its RTT (RTT_k).

That means that, CW_{nd} is increased by $\frac{RTT_k}{NIT}$ segments after each RTT_k . Then, if we consider the interval of time NIT (which must be shorter than RTT_k), we can say that CW_{nd} is incremented by one segment during this interval regardless of the value of RTT_k . However, it is not true because the sender does not receive the AKCs uniformly (i.e. the arrival processus of ACKs is not distributed uniformly). Indeed, TCP sender receives usually the ACKs in burst because the segments are sent in burst at the beginning. However, our objective is that all connections increase their congestion windows by the same size during a certain interval of time (which is greater than or equal to the longest RTT) even if the connections with short round trip time open their windows more frequently than those with long RTTs.

3 Configurations and Queueing Models

In [3], we studied CANIT performances in absence of losses for two different configurations (depicted in Figure 1). The first is composed of 5 connections, with different round trip times, sharing a single bottleneck link. The second configuration is used for study the impact of a long RTT connection which traverses a number of short RTT connections. We have considered a traditional “first-in, first-out” (FIFO) queueing scheme and gateways with infinite buffers.

In this paper, we study both of configurations in and we consider a lossy envi-

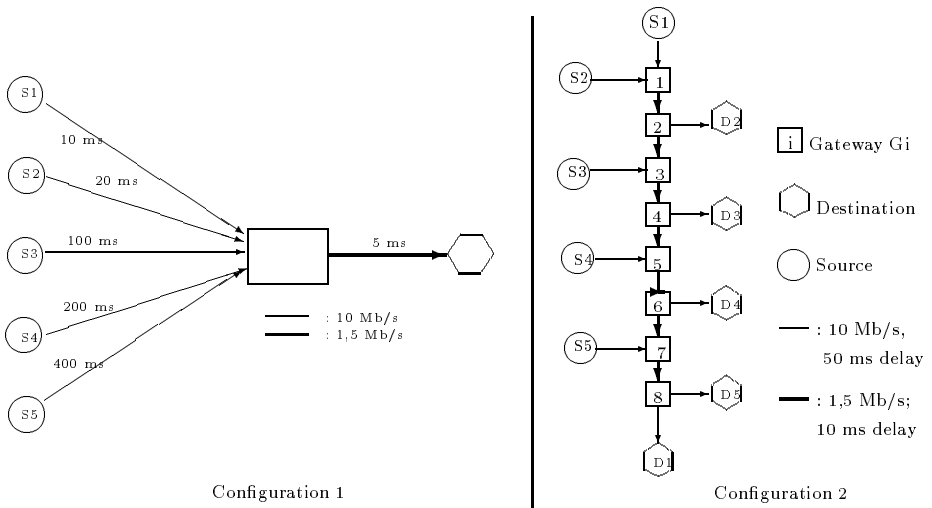


Fig. 1. Two simulation configurations

ronment in order to compare performances of the CANIT algorithm to those of standard congestion avoidance algorithm. For that, we consider gateways with finite buffer. In this case, the segments arriving from sender to a gateway are lost when the gateway buffer is full. TCP sender detects this loss when no ACK is received for a certain segment (either by a timeout or by reception of three duplicate ACKs).

3.1 The Configuration

We give in this paper only the results for configuration 1, those of configuration 2 are equivalents to those of configuration 1. This last is used to study our algorithm performances for 5 connections with different delays sharing a single bottleneck link.

Let's S_i the source of connection i , and $link_i$ ($i = 1, \dots, 5$) the link between S_i and the Gateway.

$link_i$ ($i = 1, \dots, 5$) has a delay equal to 10ms, 20ms, 100ms, 200ms, 400 ms respectively, and each $link_i$ has a capacity equal to 10Mb/s. The shared link ($link_{sh}$) has a delay equal to 5ms and a capacity equal to 1,5Mb/s.

We assume that acknowledgment path has the same delay as forward path. Then the following values of RTT_i ($i = 1, \dots, 5$) are equal to $2 \times (link_i \text{ delay} + link_{sh} \text{ delay})$

3.2 The Model

Each $link_i$, ($i = 1, \dots, 5$) is modeled by a FIFO multiple-server queue (Figure 2). The number of servers represents the capacity of the link (measured in number of segments), and time of the service represents the link delay.

S_i is modeled by a FIFO queue which holds an infinity of segments (we consider in this paper the long life connections) and the service depends on ACK received and on, of course, the used algorithm in congestion avoidance phase (CANIT algorithm or the standard one).

The Gateway is a FIFO multiple-server queue where the number of servers represents the capacity of $link_{sh}$ and the service time represents the " $link_{sh}$ " delay.

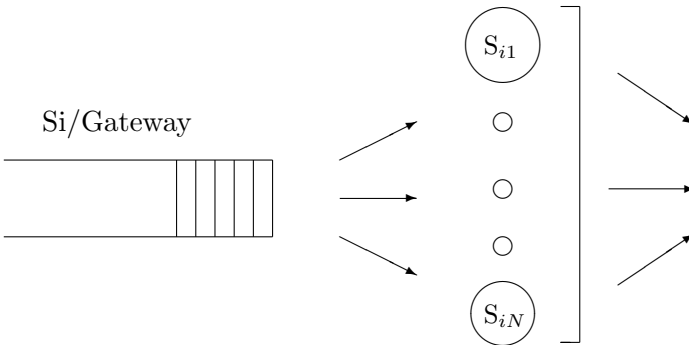


Fig. 2. Queuing Model for links of configuration 1

3.3 Performance Parameters

In order to study and compare CANIT algorithm performances to those of standard congestion avoidance one, we use the following metrics: *Fairness* and *Utilization*.

- *Fairness*: If there are N flows through a bottleneck link, each flow receiving $\frac{1}{N_{th}}$ of the capacity of that bottleneck link. We use then Jain's metric of fairness: *For N flows, with flow i receiving a fraction b_i on a given link, the fairness of the allocation is defined as:*

$$Fairness \equiv \frac{(\sum_{i=1}^N b_i)^2}{N * (\sum_{i=1}^N b_i^2)}$$

(*Fairness* = 1 corresponds to equal allocation for all users).

- *Utilization*: We define *Utilization* as the fraction -of the available bandwidth- used by connections.

$$Utilization \equiv \frac{\text{number of transfered segments} \times \text{size of segment}}{\text{rate of link} \times \text{transfer time}}$$

Here, transferred segments are original ones (retransmissions are not considered).

In [3], we show that CANIT algorithm improves successfully fairness in TCP congestion avoidance. When using CANIT algorithm, performances parameters (*Fairness* and *Utilization* are more efficient than those when using standard congestion avoidance algorithm. Moreover, when using CANIT algorithm with NIT equals to the shortest RTT of connections sharing the same bottleneck, the network is more fair and the resources utilization is more efficient. In what follows, we discuss the results concerning the lossy environment.

4 Simulation Results

This section discusses our main results obtained by simulation of the queuing model described in section 3, in presence of losses. First, Figure 3 shows the fairness behaviour of both of algorithms. TCP is more fair when using CANIT algorithm (with different values of NIT) than when using standard congestion avoidance algorithm.

Figure 4 shows that utilization (here, we consider utilization of the sharing link) bandwidth is more efficient for some values of NIT, and, when using CANIT with NIT= 30ms, both of fairness and Utilization are more efficient

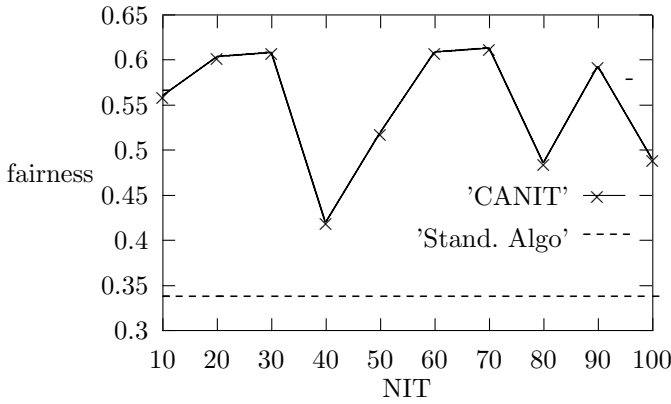


Fig. 3. fairness vs. NIT

than standard algorithm. That value represents the minimum of RTTs of the 5 connections.

In Figures 5 and 6, we use NIT equal to the shortest RTT (here =30ms). and we simulate our configuration with different values of buffer capacity. These figures show that TCP is always more fair when using CANIT algorithm with NIT=30ms and the utilization bandwidth of shared link is more efficient than in standard algorithm case.

In Figures 7 and 8, we give a comparison between the bandwidth utilization for

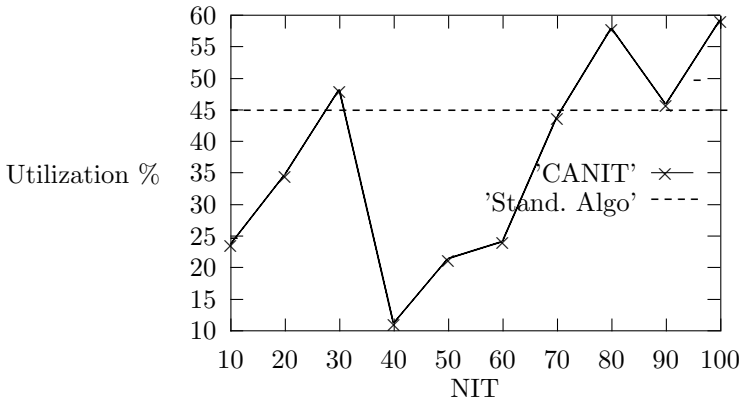


Fig. 4. Bandwidth Utilization of shared link vs. NIT.

long RTTs connections (connection 4 and connection 5). So, as showed in figures 7 and 8, for both of them, utilization bandwidth is more efficient for CANIT al-

gorithm than for standard one. This result is very important especially in TCP over satellite context.

Another important performances parameter is the loss probability. We show

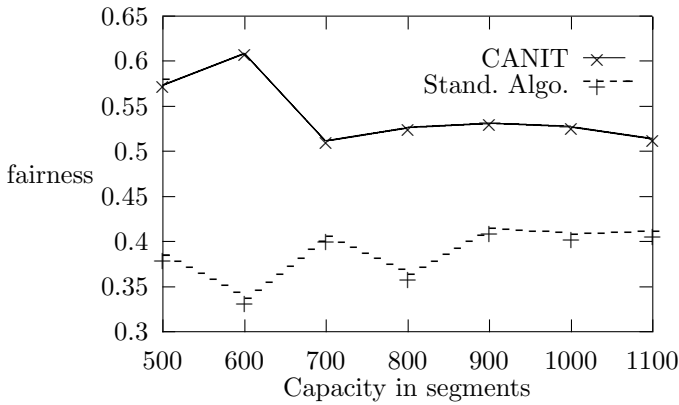


Fig. 5. fairness vs. gateway buffer capacity.

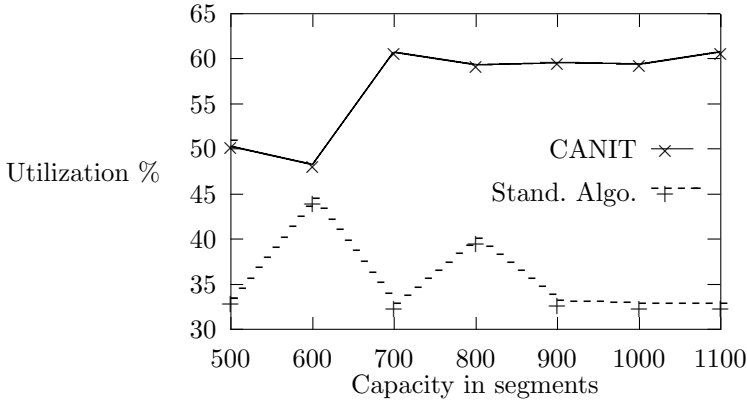


Fig. 6. Bandwidth Utilization of shared link vs.gateway buffer capacity

in figures 9 that, even if CANIT improves fairness and utilization bandwidths, the losses are equivalent to those of standard algorithm, espacially, when buffer capacities are greater than 800 segments (in this case).

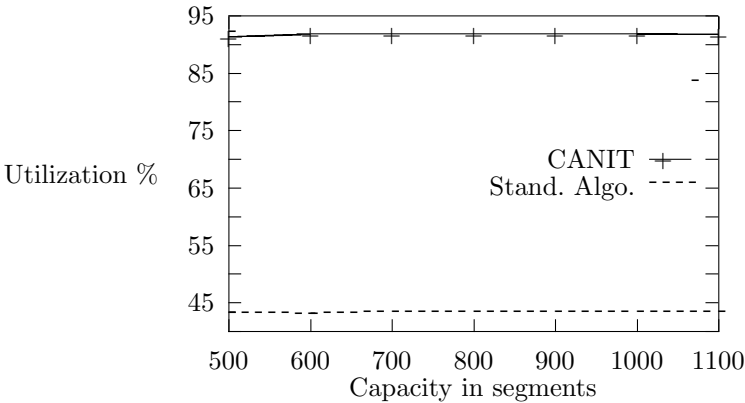


Fig. 7. Bandwith Utilization of Source4/Gateway link vs. gateway buffer capacity

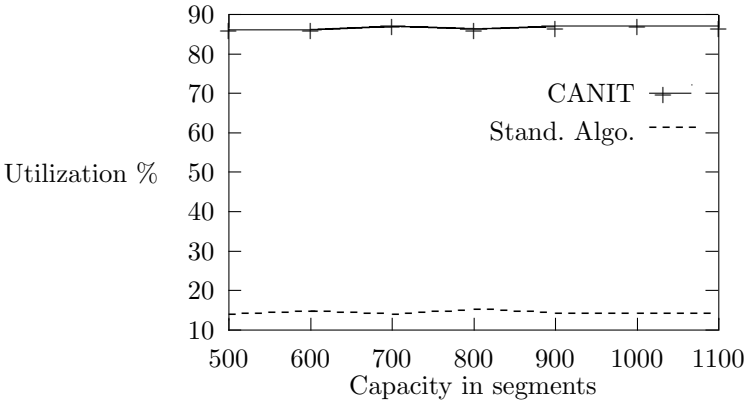


Fig. 8. Bandwith Utilization of Source5/Gateway link vs. gateway buffer capacity

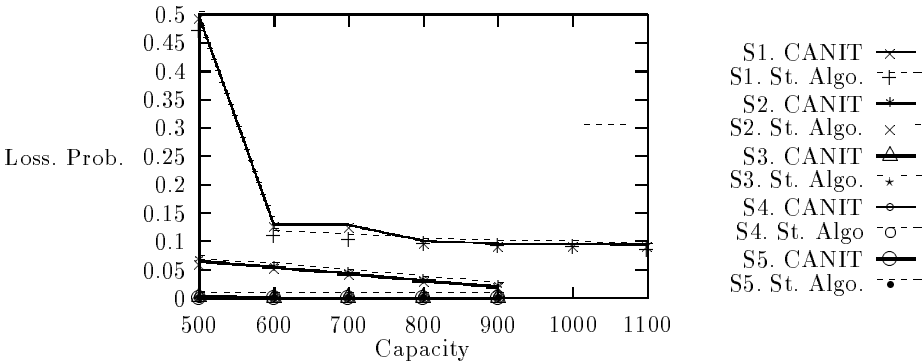


Fig. 9. Loss probability for all sources

5 Conclusion

CANIT algorithm is used in order to improve fairness in TCP congestion avoidance. We show, by simulations of queuing models that, when using CANIT algorithm in a lossy environment, performance parameters (fairness and utilization) are more efficient than those when using standard congestion avoidance. Moreover, when using CANIT algorithm with NIT equal to the shortest RTT of connections sharing the same bottleneck, the network is more fair, the resources utilization as well as the bandwidth utilization are more efficient, and the loss probabilities are equivalent compared to those generated by standard algorithm. However, implementation of CANIT algorithm requires a change of congestion avoidance mechanism at TCP sender. And, NIT estimation requires an additional mechanism. The impact of estimation time on TCP performances using our algorithm is our future work.

References

1. A. Gurtov, *TCP Performances in the Presence of Congestion and Corruption Losses*, Master's Thesis, Department of Computer Sciences, University of Helsinki, December 2000.
2. M. Allman, V. Richard Stevens, *TCP Congestion Control*, RFC 2581, April 1999.
3. H. Benaboud and N. Mikou, *CANIT: A New Algorithm to Improve the Fairness of TCP Congestion Avoidance*, Submitted in the 6th IEEE Symposium on Computers and Communications, July, 2001 Hammamet, Tunisia.
4. S. Floyd, *Connections with Multiple Congested Gateways in Packet-Switched Networks, Part 1: One-way Traffic*, ACM Computer Communications Review, V 21, N 5, October 1991.
5. T. Henderson, R. Katz, *On Improving the Fairness of TCP Congestion Avoidance*, Proceeding of IEEE Globecom'98 Conference 1998.
6. Van Jacobson, *Congestion Avoidance and Control*, Computer Communication Review, vol 18. no 4, pp 314-329, August 1988. <ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z>
7. V. Jacobson, *Modified TCP Congestion Avoidance Algorithm*, end2end-interest mailing list, April 30, 1990.
8. H. Kruse, *Performance of Common Data Communications Protocols Over Long Delay Links: An Experimental Examination*, In 3rd International Conference on Telecommunication Systems Modeling and Design, 1995.
9. J. Postel, *Transmission Control Protocol*, RFC793, September 1981.
10. B. Suter, T. Lakshman, D. Stiliadis and A. Choudhury, *Design Considerations for Supporting TCP with Per-flow Queuing*. Proceedings of IEEE Infocom'98 Conference, 1998.
11. W. Richard Stevens, *TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms*, RFC2001, January 1997.

Project Driven Graduate Network Education

A. Van de Capelle, E. Van Lil, J. Theunis, J. Potemans, and M. Teughels

K. U. Leuven, ESAT-Telemic,
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
`Johan.Theunis@esat.kuleuven.ac.be`

Abstract. In this paper projects for engineering students are described. Three different practical subjects on networking are suggested: a point-to-point link, a VoIP implementation on a LAN and an Office network. These projects are set up to offer Master students in Electrical Engineering (Telecommunications option) of the University of Leuven (K.U.Leuven) a realistic practical background. The authors consider such projects as a necessity, supplementary to the theory. With those projects, the leap to industry after graduation becomes easier to take.

1 Introduction

Most courses at university teach students only a theoretical point of view. Even in the last year of Electrical Engineering, the practical approach is sometimes left behind. After graduating, students will have to handle different problems in professional life. Theory comes in handy, but a practical solution is needed. Then, the lack of practical experience during education becomes apparent.

It is the task of a university to provide projects in which students learn to solve complex but realistic practical problems. Telecommunications students are taught courses on Networking by professor Van de Capelle. Because the authors believe a practical approach is necessary, the students also have to work out a project. The aim of this project is to offer students project skills and a strong knowledge of networking. These projects will be described in this paper.

Altogether, there are 27 students. These students are divided in three major groups. Each group has a different project to manage. The subjects are sketched in Section 2. Inside each group, there are several small subgroups, whose mission slightly differs.

Each group first has to analyse the given situation. Then, they have to perform the appropriate measurements of traffic load. Meanwhile, an analytical model of the configuration has to be invented. Starting from the measurements, the students simulate the configuration in OPNET and compare the outcomes with these of the analytic modelling. With this knowledge, the students predict the consequences of implementing a new technology or service. Finally, a concrete realisation makes the project complete.

In a first section we will focus on the different subjects proposed. Afterwards the supervision is described. After this, results of the students are discussed. Finally, the realisations inside the projects are pointed out. Further information can be found on the website of the projects [1].

2 Outline of the Subjects

Three different subjects are treated: a point-to-point connection, a LAN-PABX and an Office LAN.

2.1 Point-to-Point Connection

The K.U.Leuven consists of two geographically separated campuses: one in Leuven (K.U.Leuven) and one in Kortrijk (KULAK). These two campuses are interconnected with a 2 Mbps full duplex leased line. At this moment, only a small part of the leased capacity is used.

Because of this inefficiency, the University of Leuven plans to use the remaining capacity for phone traffic. This way, phone calls to KULAK will be for free, while external calls to zone Kortrijk and adjacent zones will be at local rate. The other way around is also possible: employees of the KULAK will be able to call their colleagues of the K.U.Leuven for free while external calls to zone Leuven and neighbouring zones will be at local fee. Figure 1 illustrates the above.

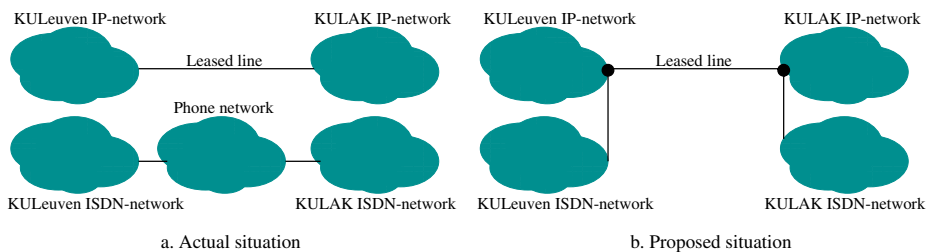


Fig. 1. The K.U.Leuven-KULAK case

Besides voice, the K.U.Leuven also wants to use the link for video conferencing and dial-up networking: courses and lectures can be transmitted as video broadcasts, meetings can take place from remote distances and the surroundings of the KULAK campus can connect to the K.U.Leuven IP network at local call rates.

It is up to the students to examine this case and to find out what is (im)possible. They also have to choose between two possible ways of transporting the voice traffic: by circuit switching or by packet switching. They have to consider the differences between the ISDN standard and other codecs [2], paying special attention to capacity demand, packet loss and delay.

2.2 LAN-PABX

In most residences of the K.U.Leuven, students have a LAN connection in their room. This LAN is typically a shared Ethernet [3], common for the whole residence. In the bigger residences, the LAN is a partly switched Ethernet [3]. The

LAN is connected to the Internet. There are also collective phones, enabling students to make internal as well as external phone calls. Of course the LAN and phone network are separate infrastructures.

Today, it is no longer needed to keep these two networks separate. In the LAN-PABX concept the LAN is also used for phone traffic. Computers with a microphone and speakers replace phones. The PABX is replaced by a server taking care of the signalling. Another server, the gateway, performs the conversion to the external phone lines.

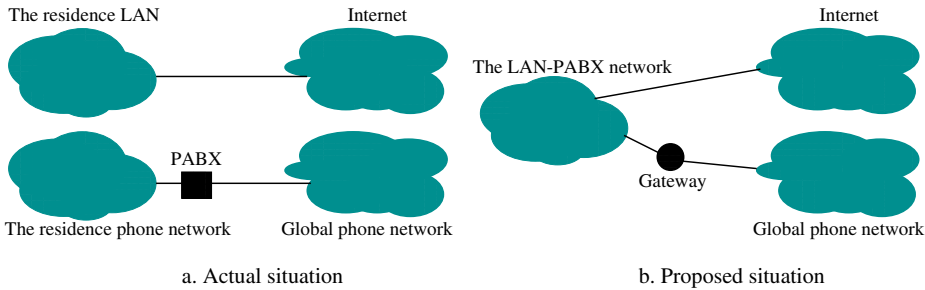


Fig. 2. The LAN-PABX case

The K.U.Leuven plans to offer students living in residences the services of such a LAN-PABX. Of course, a study is needed to provide enough information about possible problems and bottlenecks. Starting from the data traffic on an existing residence, students have to examine whether VoIP on the LAN can be provided or not. If so, an estimation of the offered quality for voice and the drawbacks for the existing data traffic has to be made. Figure 2 shows the architecture of a separate LAN and PABX and the one of a LAN-PABX.

The gateway between IP and ISDN can be located on site (in the residence itself) or in a central point (where the connection between the K.U.Leuven IP network and the Internet is made). In the first case the voice traffic only loads the LAN, in the latter case voice traffic has to travel through the whole network. The qualitative and economic differences between those two strategies have to be examined as well.

2.3 Office LAN

A typical application area of the LAN today is the office environment. Workstations, servers, x-terminals, PCs, printers etc. collaborate using the LAN. As an example, the students have to look at the office LAN of ESAT-Telemic, the research group to which the authors adhere. This project focuses on the application layer protocols that are used for file serving across the network: either the Networking File Serving protocol (NFS) [4], either the Server Message Block protocol (SaMBa) [5] can be used to share physical hard disks among different

computers. This way, the directory structure presented to the user is independent of whatever host he is actually working on: NFS or SaMBa are used to transport the data from the hard disk where it is stored to the computer where it is required, as if it was stored on a local hard disk.

It is up to the students to redesign the file sharing on the ESAT-Telemic LAN in an optimal way. This redesign is not straightforward. First of all the workstations can be used as file server and as application server: the workstations can be used to store the user's data and can be used to run applications. Both functions can be separated or combined: in the former some workstations are used as file server and other as application server exclusively, in the latter option both functions are combined on a workstation. Subsequently the selection of a file serving protocol is required. This has consequences regarding performance, security, service guarantee, fault redundancy etc. If SaMBa is the selected protocol, then one or more SaMBa server(s) need to be assigned.

Students have to calculate and simulate performance indicators, such as delay and network load, for several possible network architectures. After a comparison, the optimal office LAN configuration can be determined.

3 Supervision

The students were supervised by a team of three research assistants, each being responsible for one part of the project: measurements, analytical modelling and simulations. A computer classroom was reserved one afternoon a week to work on the project. After an introductory session, every week a different part of the project is dealt with, alternating between measurements, analytical modelling and simulations.

During these afternoons, a group discussion for every project showed the general progress and inspired students to co-operate. After the three discussions, they were able to ask questions about the specific topic. The afternoon sessions were certainly not enough to finish the project. A lot of work had to be done at home. In case of problems, students could contact the responsible assistant by email or simply go to his office. For the realisations, the team of assistants was reinforced with two experts sharing their experience with the students.

4 The Approach of the Students

In this section the approach of the students towards fulfilling the projects is given. Measurements, analytical modelling, simulations and realisation of each project are discussed separately.

4.1 Point-to-Point Connection

Measurements

Data Traffic. Of course, the actual data traffic on the leased line has to be measured. Therefore, the routers can be contacted using **SNMP** [6]. This method is based on the fact that the two routers on each side of the leased line keep track of the total number of bits and packets processed. With a tool like **snmpget** [7] in Unix or Linux, information about the amount of incoming and outgoing packets and bytes, busy hour, the number of discards and the speed of the interfaces can be retrieved.

Figure 3 illustrates the leased line between K.U.Leuven and KULAK. Measurements are performed by polling both *cisco-kulak* and *cisco-kulnet* regularly. The routers return on each SNMP poll the accumulated number of processed bytes and packets. By subtracting two subsequent measurements, the number of processed bytes and packets in the interpolling interval can be retrieved. These measurements were automated using Perl [8] and lasted for 11 days (no week-ends included).

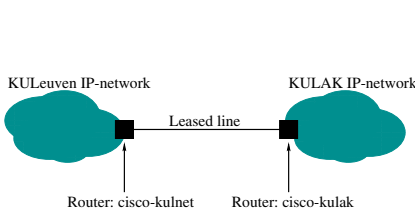


Fig. 3. Measurements on the leased line

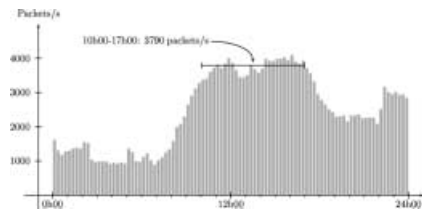


Fig. 4. Packets arriving at *cisco-kulnet*

The measurements afterwards were condensed in easy-to-interpret figures. Figure 4 is an example. It shows the average number of packets arriving at *cisco-kulnet* during a day. Out of the graph, the busy period can be extracted. As can be seen on Fig. 4, the busy period starts from 10h00 and lasts until 17h00. Measured values during this busy period will be used instead of all the values. This is quite obvious: during this period the leased line will suffer the most from the added voice traffic.

For other nice graphs, we refer to the website of the projects (follow the link to the reports) [1]. The obtained values and characteristics are needed for the analytical modelling and simulation.

Phone Traffic. Calling information is also needed. The number of calls from K.U.Leuven to KULAK and vice versa, a profile of the call duration and busy hour has to be extracted from the call records of the K.U.Leuven. In these call records, each outgoing call is registered, along with the source, destination and duration.

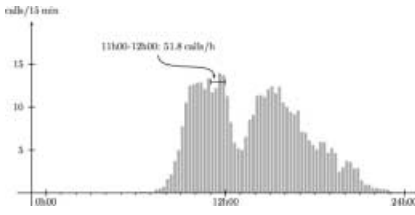


Fig. 5. Call frequency from K.U.Leuven to KULAK

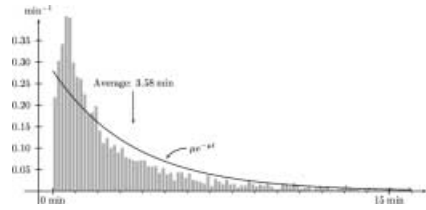


Fig. 6. Probability density of call duration

Some results obtained by the students are given in Fig. 5 and 6. In Fig. 5, the frequency of calls from K.U.Leuven to KULAK is shown. Out of this graph, a busy hour can be extracted. Figure 6 illustrates the probability density of the call duration. In this figure, an exponential distribution curve is fitted to the measurements. This results in two values: λ and μ , both needed in modelling as well as in simulations.

Analytical Modelling. Voice applications put high restrictions on the total end-to-end delay. This delay mainly consists of a propagation time, a coding-decoding delay in the gateway and queueing delays in the network. Depending on the source, the maximum tolerable delay for an interactive voice communication ranges from 150 to 250 ms.

To save bandwidth students did not use the G.711 protocol on the leased line (as used in the ISDN standard), but the G.723.1 codec resulting in a data rate of 5.3 or 6.3 Kbps. If only one voice frame is included in a single packet, the bitrate on IP level increases to 16 Kbps (including the RTP-, UDP- and IP-header). Each voice frame has a duration of 30 ms, which gives rise to a packet rate of 33 p/s. In this case, the coding-decoding delay in the gateways amounts to 85 ms.

The propagation time is less than 1 ms and can therefore be neglected. Most of the effort is spent on the analysis of the queueing delays in the routers (*cisco-kulnet* and *cisco-kulak*).

A simplified model for a router is depicted in Fig. 7. The router consists of a central queue in which all packets arrive and an outgoing queue for each interface. The speed of the central queue represents the routing capacity expressed in packets/second. It is hardware specific: for the *cisco-kulnet* router it is 100000 p/s and for the *cisco-kulak* router 40000 p/s. The speed of the interface queue depends from the type of interface and is expressed in bits/second. Our leased line has a capacity of 2 Mbps. The gateways have an Ethernet connection to the routers.

For simplicity, arrival times can be taken exponentially distributed. Buffer capacity is assumed to be infinite. No packet loss can be modelled this way. An M/D/1 queue can be used to model the central route engine queue: an average address lookup takes the same amount of time regardless the packet size. A

switch on the other hand requires an M/G/1 queue. The service time depends on the packet length.

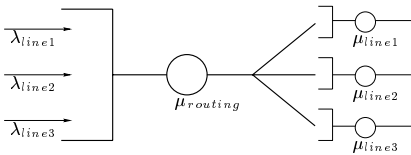


Fig. 7. A simple model for a switch or a router

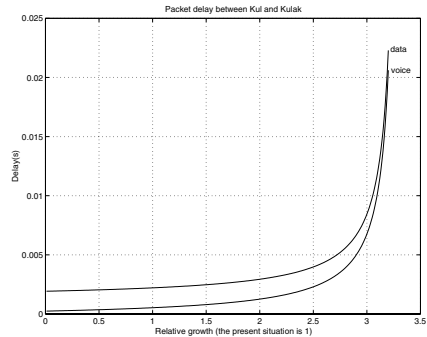


Fig. 8. Total packet delay from K.U.Leuven to KULAK

Figure 8 shows the total delay induced in the *cisco-kulnet* router as function of the number of employees of the K.U.Leuven campus (a scale relative to the current situation is used).

Students also calculated the delay values in case voice packets have priority over data packets. The difference in queueing delay becomes clear in Fig. 9.

A gateway can deal a limited number of simultaneous calls. The probability of blocking, i.e. when all lines are busy, can be modelled by an Erlang B distribution. The maximum allowed blocking probability determines the minimum number of trunks (and thus gateways) needed. The blocking probability for the current phone load of the K.U.Leuven can be found in Fig. 10.

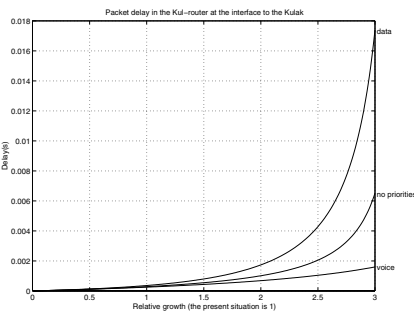


Fig. 9. Interface queueing delay from K.U.Leuven to KULAK

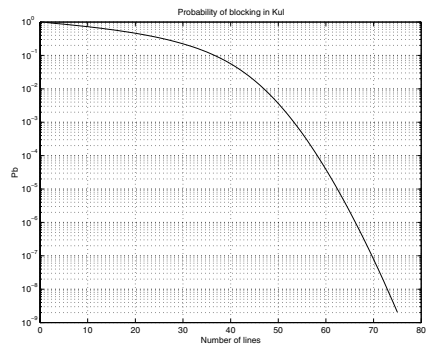


Fig. 10. Gateway blocking probability for current K.U.Leuven phone load

Due to space limitations we cannot give all results here. For more information, we refer to the website of the design projects [1].

Simulations. The same holds for the simulations as for the measurements and the modelling: also here we can only give some results. For more detailed information, we refer to the website of the projects [1].

For simulating the configuration, OPNET [9], was used. Reality had to be imitated as accurate as possible. An example of such a network topology in OPNET is shown in Fig. 11.

Once running properly, performance parameters roll out of the simulations. These parameters, such as traffic load, delay, packet loss and available capacity, can be compared to the measured and modelled values. An example of plots resulting from OPNET is given in Fig. 12. The figure shows the queueing delay between *cisco-kulnet* and *cisco-kulak*.

4.2 LAN-PABX

Measurements. The traffic on the LAN of the residence has to be recorded. Therefore, `tcpdump` can be used. Transmission capacity, a useful parameter when implementing VoIP, can be measured with `netperf` [10]. In case the gateway is located at a central point, the data traffic on the K.U.Leuven IP network has to be included as well. Also an estimate has to be made of the voice traffic that will be generated by a student on a residence.

In this limited amount of space, we can only give some of the results. For more detailed information, we refer to the website of the projects [1].

On the LAN, the students measured the first 10,000 packets in each hour, this for one week. Out of these measurements, they extracted the busy period. This period is shown in Fig. 13. As can be seen in the figure, the 10 Mbps Ethernet is almost saturated at the busy period.

Another interesting graph is the one shown in Fig. 14. This figure shows the distribution of the packet sizes as they were recorded on the LAN. These outcomes can be imported directly in OPNET.

The measured data can be split up into different classes according to the applications which generate the data (ftp, mail, www, ...). Again, we refer to the website for more information.

Using `snmpget` on the routers in the K.U.Leuven IP network, figures similar to the ones in Section 4.1 can be constructed. The principle is the same, so we will not go any deeper into detail.

Analytical Modelling. The voice application is no longer used in a simple point-to-point environment, but on a shared Ethernet. As stated before, there is a maximum acceptable delay to maintain a good quality.

Also here, the G.723.1 codec is used to save bandwidth. Students on the residence can either make a phone call with an IP-phone or with a software tool installed on their computer. Both act as a gateway. The PC's soundcard together

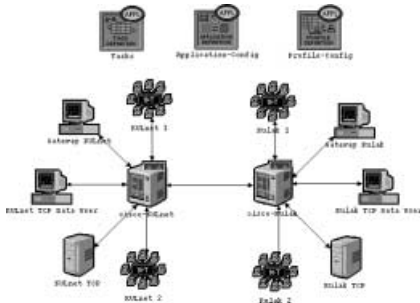


Fig. 11. A network topology in OPNET

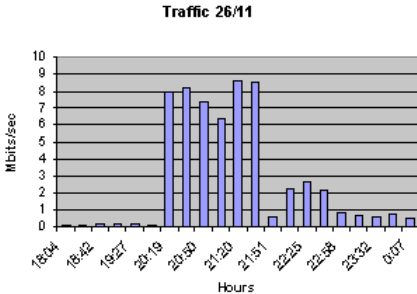


Fig. 13. Busy period on the LAN

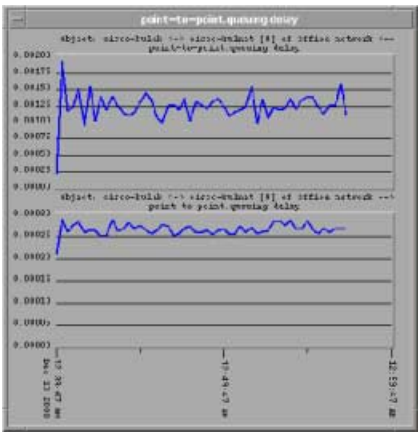


Fig. 12. Queueing delay between K.U.Leuven and KULAK

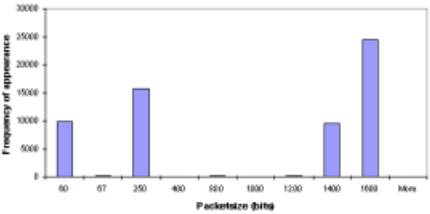


Fig. 14. Packet size distribution on the LAN

with a microphone causes about 30 ms of delay. The encoding-decoding delay still is 85 ms.

For internal calls (on the same subnet) the network delay is caused by the Ethernet. For external calls an additional delay is induced by other network elements, such as routers.

The delay and the packet loss on the shared Ethernet can be calculated using one of the two models described in [11]. They were the only practically useful models, including the truncated binary exponential back-off algorithm, we found in literature. Both models assume a Poisson process for the arrival of frames and the total arrival of frames and retransmissions. In the more complex model of the two, the collision probability of a frame is dependant of its current back-off state.

The measurements show that most of the traffic is generated by a limited number of users. Only 4 heavy users are counted during a busy hour. The Ethernet delay as function of the number of heavy users is depicted in Fig. 15.

It is clear that the subnet is near its maximum capacity. An upgrade to a 100 Mbps or a switched Ethernet is necessary to implement the voice service in the residences.

In case of external calls, the delay of other network components can be calculated in the same way as described for the point-to-point project. More information can be found on the website of the design project [1].

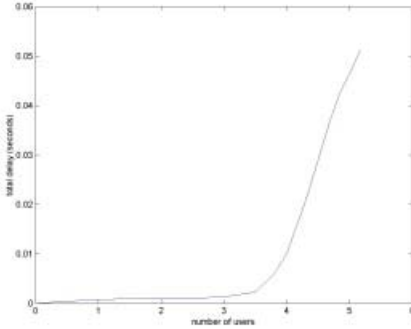


Fig. 15. Ethernet delay as function of the number of heavy users

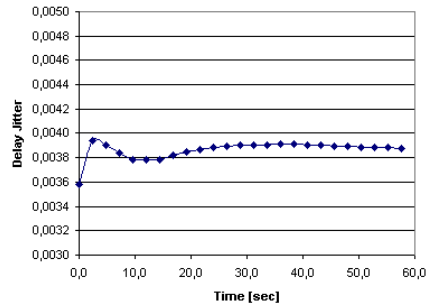


Fig. 16. Delay jitter during a simulation

Simulations. As in Section 4.1, this configuration was simulated in OPNET [9]. Again, different performance parameters can be extracted out of the simulations. As an example, we give the evolution of the delay jitter during a simulation in Fig. 16. In this simulation the measured amount of traffic on the LAN is generated along with supplementary voice traffic. This simulation gives an approximation of reality when VoIP would be implemented on the LAN.

Other plots can be found on the website of the projects [1].

4.3 Office LAN

Measurements. A first set of measurements is performed to obtain the required data on the load and use of the network file service. Out of these measurements, the common profile of a user on the ESAT-Telemic network is extracted. This profile includes the number of files a normal user requires in one hour, the distribution of the interarrival time and the size of these files. The profile is also refined by looking at applications in the measured traffic. The amount of traffic generated by different applications (www, ftp, nfs, X11, smb, ...) is determined. The profile can be found in the reports, available on the website of the projects [1].

The total load of the office LAN has to be measured as well. Therefore, `tcpdump` can be used. Different characteristics of the data traffic can be determined: the average load on the LAN, the peak load, packet size distribution,

interarrival times of the packets, ... As an example, Fig. 17 shows the packet size distribution of the packets measured on the LAN during busy hour.

Next to the load measurements, the performance of the workstations needs to be analysed as well. Here the difference between an old and a newer workstation should be taken into account. In an optimal scenario the newest and quickest workstations should be used as application and file servers, the older ones can be protocol servers (e.g. SaMBa servers).

Table 1. Time to copy one byte of an average file and speed multiplier

Workstation	Time (<i>ms</i>)	Speed Multiplier
Hercule	1.29	14.35
Toine	5.17	3.59
Blondine	5.43	3.42
Loebas	18.58	1.00
Kastaar	15.38	1.21
Duchesse	6.51	2.85
Zulte	16.16	1.15

Table 1 shows the results of such a measurement. In the first column, the different workstations of Telemic are summed. In the next column, the times to copy one byte of a file of average length for each workstation are indicated. As you can see, Hercule is the fastest workstation and Loebas is the slowest one. In the third column, relative speed multipliers are accorded to each workstation. The slowest workstation, Loebas, is accorded the multiplier 1. The multiplier indicates the times a workstation is faster in copying files than Loebas. More results can be found on the website [1].

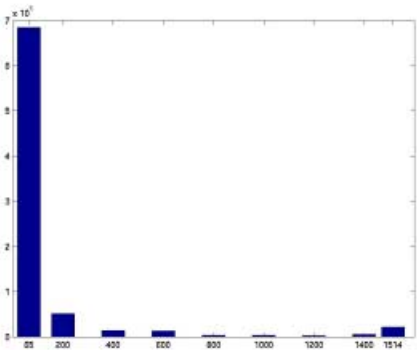


Fig. 17. Packet size distribution on the office LAN

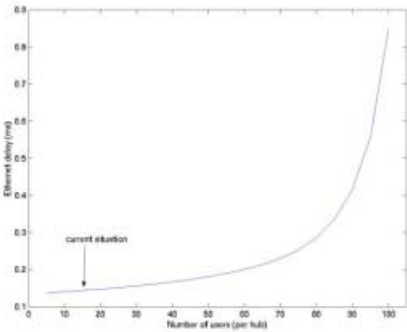


Fig. 18. Ethernet delay in function of network utilisation

Analytical Modelling. The file serving delay consists of two parts: the network delay and the server delay.

The network delay mainly consists of the delay in the shared Ethernet. It can be calculated as described above in Section 4.2. Figure 18 shows that the network utilisation is low and the delay is very small. The network is clearly not causing any trouble.

The number of file transfers (differentiated into small, medium and large files) per time unit, together with their service times should give enough information to calculate the server delay.

A file server can be modelled by an M/G/1 queue, either receiving complete files or the separate packets used to transfer a file. This difference between units has its consequences. In the former case simultaneous file transfer is impossible but a simple combination of M/D/1 queues offers very quick results. The delay of the file server is much higher than the network delay and will therefore be the bottleneck of the system. The latter case better fits reality but interpretation of the results is far more complicated. Now a single file consists of multiple packets. The total delay is not equal to the mean packet delay multiplied by the number of packets. It is not clear how to calculate the delay of a file when its packets are mixed with other's.

For a combined server the picture becomes even more complicated. Not only files or packets arrive in the queue but also elementary tasks. These tasks originate from applications running on the server. Measurements in a controlled environment make it possible to estimate the elementary task rate for a specific application and the service time for such a task.

This reasoning however is not correct. When multiple applications run at the same time, they are influenced by each other and produce less elementary tasks per time unit as a result. It is not clear yet how to model this feedback.

Simulations. OPNET [9] was used to simulate these configurations. The same parameters as in Sections 4.1 and 4.2 can be retrieved from the simulations. Results can be found on the website of the projects [1].

5 Realisation

Because it is important to visualise the problems and solutions, a kind of realisation is required. At ESAT-Telemic, there is a network lab specifically intended for such purposes. The lab consists of several computers and some routers and hubs.

The group handling a point-to-point connection had to rebuild the actual situation in the lab from scratch. This helped them to learn what a point-to-point connection is all about.

The Office LAN group had to build a small LAN and get some servers and clients running, using either NFS or SMB. Afterwards a protocol trace was taken on both SMB and NFS. This made them understand the principles of file sharing.

The group of the LAN-PABX on the other hand had the opportunity to help with a real implementation of VoIP. With the support and material from Siemens, a temporal LAN-PABX was installed on a residence of the K.U.Leuven. Of course this was a great help in understanding the VoIP concept.

6 Conclusions

In this paper, projects are proposed for students who follow a graduate course on networking. The lack of practical insight after such courses is compensated by solving a practical problem. The proposed projects are close to reality; in fact the K.U.Leuven can use the outcome of the work of the students for future decisions.

Acknowledgements

Simulations were performed with OPNET licenses in the framework of the OPNET University Consortium.

We would also like to thank the Telecommunications students, who spent many hours on the projects in order to get some nice results.

References

1. "Home Page of the Network design project", <http://www.esat.kuleuven.ac.be/~3irtele/H239/>
2. "Homepage of ITU", <http://www.itu.int/home>
3. A. S. Tanenbaum, "Computer Networks", Prentice Hall, 1996, pp. 276-287. ISBN 0-13-394248-1
4. S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C. Beame, M. Eisler, D. Noveck, "NFS version 4 Protocol", RFC 3010, December 2000
5. "Home Page of SaMBa", <http://www.samba.org>
6. J. Case, M. Fedor, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)", RFC 1157, May 1990
7. "The NET-SNMP Home Page", <http://net-snmp.sourceforge.net>
8. "Perl Home Page", <http://www.perl.com/pub>
9. "OPNET Home Page", <http://www.opnet.com>
10. "The Public Netperf Homepage", <http://www.netperf.org/netperf/NetperfPage.html>
11. J. Parry, "Two Models of Ethernet Networks", Metron Technology Ltd, September 1997

Modelling User Interaction with E-commerce Servers

Hubert Graja and Jennifer McManis

Performance Engineering Laboratory, School of Electronic Engineering
Glasnevin, Dublin 9, IRELAND
{grajah, mcmanisj} @eeng.dcu.ie

Abstract. E-commerce is an increasingly significant part of the global economy. Users of E-commerce Web sites often have high expectations for the quality of service, and if those expectations are not met, the next site is only a click away. A number of performance problems have been observed for E-commerce Web sites, and much work has gone into characterising the performance of Web servers and Internet applications. However, the customers of E-commerce systems are less well studied. In this work we seek to quantify customer satisfaction with a Web site. We observe that customers may be categorised either based on their satisfaction ratings, or on other factors (in which case satisfaction can be assessed for the customer class). Web sites may then be evaluated relative to the different customer categories, with potentially more care being given to the satisfaction of high priority customers. We present a methodology for deriving customer satisfaction, and apply it to the evaluation of two academic Web sites.

1 Introduction: E-commerce Customer Satisfaction

The World Wide Web (WWW, or Web) is one of the most important Internet services, and has been largely responsible for the phenomenal growth of the Internet in recent years. An increasingly popular and important Web-based activity is E-Commerce, in which various types of financial transactions are carried out or facilitated using the Web. It is widely expected that E-Commerce activity will continue to grow and be a significant component of the global economy in the near future. In an area such as E-Commerce, users often have definite expectations about the service they receive (based on similar non-Internet transactions) and/or about Web service in general (based on previous Internet usage). Therefore, E-Commerce users typically demand high quality service, such as: desired information is easy to find, accurate, and current; the retrieved information is of good quality; establishment and download delays are low; and there is high availability of the Web server.

A number of performance problems in E-Commerce systems have been observed, mainly due to heavier-than-anticipated loads and the consequent inability to satisfy customer requirements. E-Commerce systems have often been designed under tight delivery schedules and without due regard for the impact of different possible loadings on the performance of the system. This has resulted in a lot of work attempting to characterise the performance of Web servers and Internet applications e.g. [1], [2], [3], [4]. However, the customers of these E-Commerce systems are less well studied, despite the fact that some surveys show considerable dissatisfaction with

current E-Commerce and Web servers. For example, it has been reported that as many as 60% of users typically cannot find the information they are looking for in a Web site, even though the information is present [5]. Similarly, customer dissatisfaction with WAP phones is attributed to difficulties in accessing Internet applications [6], including low access speeds and the large number of menus traversed to access information. In an area such as E-Commerce, customers demand high quality service, and it is easy for them to move away to another site if they perceive the current one to be unsatisfactory.

Determining how well customer requirements are met for all users of a Web site is difficult, and is often based on unrepresentative sampling or anecdotal evidence or on the experience of a “generic” user. By quantifying the factors involved in these requirements, it is possible to analyse and predict customer satisfaction with an E-Commerce sit in a more scientific manner. Customer satisfaction may be quantified using various parameters, such as *response time*, *number of clicks* needed to find the desired information, *amount of information* the customer is required to give, *quality of information* they sent, *security* of the interaction, and *predictability* of the service received. It is entirely possible that different users will attach different satisfaction levels to the parameters and also that they value the satisfaction of some parameters over others.

Since customer satisfaction is critical to the success of E-Commerce systems, characterising the customer’s requirements is an important consideration in designing such systems. Our approach involves:

- Constructing a customer model which captures the satisfaction that the customer gets when using the site;
- Dividing the customers into distinct categories based either on how they judge their satisfaction, or on some other parameters.

Web server performance may then be assessed relative to the different customer categories. Those parameters that the customers value most should be optimised first, and if certain categories of customers are more “valued”, the server can be geared to maximising those customers’ satisfaction level. Potentially, servers may even be designed to serve different customer categories differently.

In the following we discuss customer categorisation and satisfaction. We present a methodology for quantifying customer satisfaction. Finally, we demonstrate our methodology in the assessment of customer satisfaction for two Web sites.

2 Customer Satisfaction and Categorisation

Our goal is to quantify customer satisfaction with a Web site in a uniform way. To do this we must be able to measure customer satisfaction according to the assessment of various parameters. This measurement must be mapped to some fixed satisfaction scale (e.g. 1 to 10 with 10 being the best). Finally, categories of customers must be defined which have similar satisfaction measures.

Some of these parameters may be measured in an objective manner. For instance, response time can be measured in seconds. Similarly, the number of clicks to reach a given piece of information can be counted. Thus only the task of mapping the measurements onto the satisfaction scale remains. Other parameters are not as easy to

measure objectively. For instance, how does one determine the ease or difficulty of navigating a Web site to find specific information? What one customer sees as simple, another may find confusing and unmanageable. Customer surveys might help to determine good measures for these less easily measured parameters. Even for parameters with objective measurements, mapping to a fixed scale may be done in various ways. For instance with response times, one customer may be willing to wait one minute, but no longer, while another becomes gradually more dissatisfied with waiting.

Different customers may assign different satisfaction levels to different parameters. For instance, one customer may decide they are willing to give information such as name, address and telephone number to a Web site, while another may find it unacceptable to give any information at all. In addition, different customers may apply different weightings to the same parameters in judging their overall satisfaction with a Web site. At one extreme, this includes the case where one or more of the parameters gets a zero weighting, e.g. if all the customer cares about is response time, all other parameters are zero-weighted. Combining these parameters in order to quantify overall satisfaction leads to *categories of customers* with similar satisfaction measures and similar weightings.

Ideally, customer categorisation classifies all customers based on how they judge their satisfaction with an E-Commerce system, grouping together those with similar requirements. In practice, it is not possible to determine the customer satisfaction for each individual customer. Instead, customers can be divided into distinct categories in some other way such as large/medium/small budget; type/speed of Internet connection the customer has; or frequent/previous/new customer. Customers in distinct categories may very well have similar satisfaction requirements (at least for some satisfaction measures). For example, if customers are divided according to means of access, WAP phone users might be willing to tolerate longer delays than PC-based users, as well as preferring a low number of hops to access information.

3 A Methodology for Quantifying Customer Satisfaction

The first step in our approach to assessing Web site performance via customer satisfaction is defining a **customer list** C consisting of one or more customers and a **parameter list** P consisting of features of a Web site which will potentially affect customer satisfaction. For each customer in C , their behaviour is defined in terms of their interaction with the Web site. A **trace behaviour** for a customer is defined as the series of clicks and other information that the customer exchanges with the site. Typical behaviour for a customer or customer may then be defined as one or more traces and satisfaction is measured relative to these trace behaviours. By further associating customers with customer classes we may arrive at a satisfaction measure for customer classes as well.

For any trace behaviour, a measure of customer satisfaction relative to the parameter list can be defined as follows:

- 1) For each parameter $p \in P$, define a quantification of satisfaction Q_p . For instance, if p is the number of clicks, Q_p is easily defined as an integer value. Other parameters may have more subjective quantifications. For

instance, how does one quantify the "quality" of information available at a Web site?

- 2) For each customer/parameter pair $(c,p) \in C \times P$, let $M_c: Q_p \rightarrow \text{Scale}$ map the quantification of satisfaction to a fixed range of values $\text{Scale}=[\min, \max]$. This mapping allows a large number of parameters to be compared in a uniform fashion. Note that **Scale** may take on either discrete or continuous values. Even in the case that a parameter is easily quantifiable (as in the number of clicks), the mapping of this to **Scale** may be subjective according to customer class. A WAP phone user may find it unacceptable to make 10 clicks to reach some information, while a PC-based user may find this tolerable.
- 3) For each customer, an n -dimensional satisfaction vector may now be defined (where n is the number of elements in the parameter list) incorporating the customer satisfaction for all parameters. This satisfaction vector can be thought of as the "raw data" for determining customer class satisfaction.
- 4) Different customers may value Web site parameters differently. For each customer class, determine a satisfaction weighting for each parameter i , denoted $W_c(i) \in [0,1]$ where

$$\sum_{i=1}^n W_c(i) = 1$$

These weightings are used along with the satisfaction vector from step (3) to define an overall satisfaction level for each customer trace.

Thus for each behaviour trace we have arrived at a satisfaction measure for each individual parameter and for the Web site as a whole. By defining customer class as a collection of trace behaviours, we may extend our satisfaction measure to customer classes.

- 5) Let a **customer class** be defined as a collection of trace behaviours. The class satisfaction measure is defined as a weighted sum of the trace satisfaction measures. (It may be considered that some behaviour is exhibited more frequently by a user in a Class, and this behaviour should be given higher weighting).
- 6) Finally a weighting of customer classes can be defined, allowing for an overall satisfaction measure for the Web site. By varying this weighting, we can study how favouring certain customer classes over others affects overall customer satisfaction with the site.

The most difficult part of this exercise is in relating customer trace behaviour to the satisfaction vector. How parameter satisfaction is measured and how it is mapped onto the Scale must be addressed on a case-by-case basis, although experience using the methodology may lead to the definition of some standard cases. Also, since multiple executions of the same trace may lead to different values, some statistical analysis may be required. An overview of the methodology is given in Figure 1.

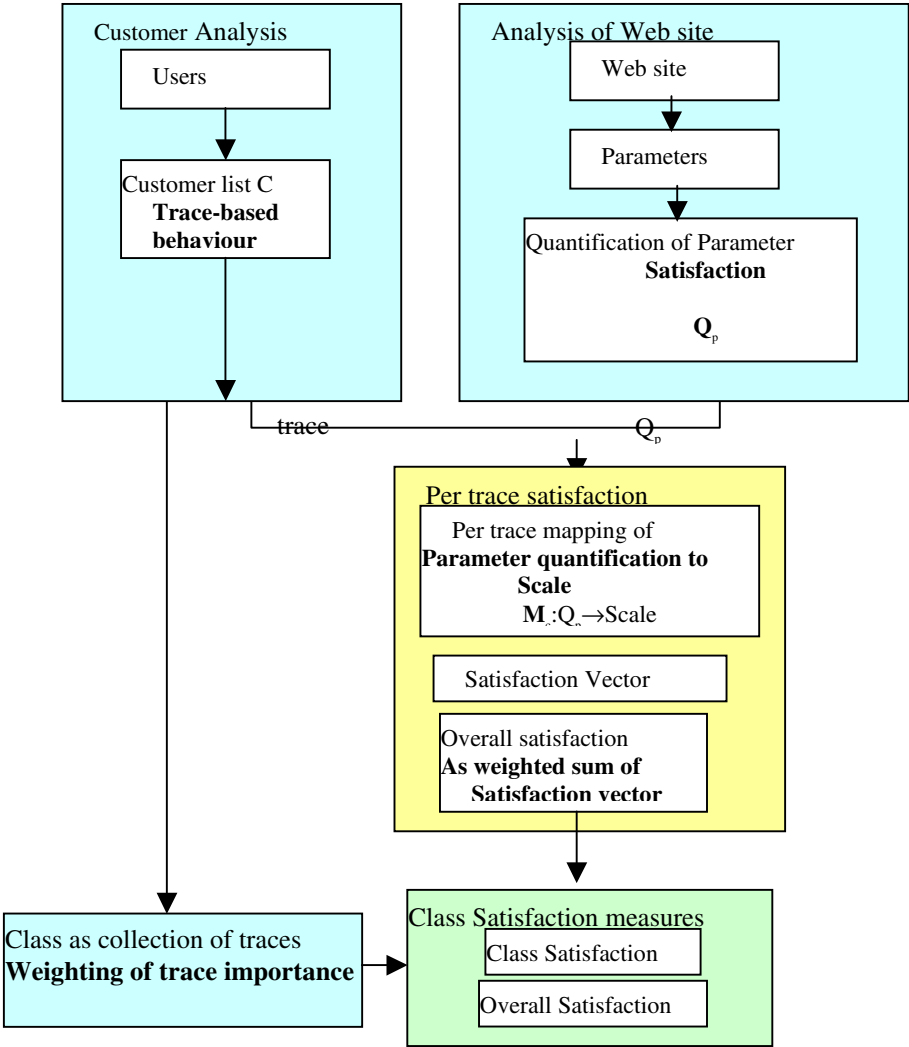


Figure 1: Overview of Methodology

4 Application of the Methodology

The approach outlined in the previous Section has been applied to the analysis of two university Web sites – Dublin City University School of Electronic Engineering [7] and Florida International University [8]. For each of these sites, four classes of users were defined: internal students, external students, internal staff, and external staff. These user classes were distinguished by their behaviour, which in all cases was to seek some relevant information. For each user class, multiple traces were specified and a path weight assigned to each trace indicating its relative usage.

Three Web site parameters were defined: *complexity*, *time*, and *quality of information*. The quantification and assignment of satisfaction values was the same for all user classes. Complexity was quantified as the number of clicks in a trace; time was quantified as the overall time to complete a trace; and quality of information was judged subjectively based on how close the information retrieved was to what was actually sought. The scale chosen was a simple 0-to-4, with 4 being best and 0 worst. Complexity and time were divided into bands, and each band assigned a value in the scale; quality of information was assigned a value based on a survey of ten users.

Data was gathered using the Web Performance Trainer 2.1 tool [9] to execute each of the traces on the actual Web site in question. This was necessary solely to take time data, and was carried out both on a weekday and a weekend day. The other two satisfaction values were determined by an inspection of the Web site. Tables 1 and 2 in Appendix 1 summarise the data for the DCU and FIU Web sites, respectively. Analysis of the results and what conclusions we can draw from this study are given below.

5 Conclusions and Future Work

Modelling customer satisfaction with Web and E-Commerce sites is not as well studied as Web server modelling, but determining whether and how the customers of these sites are satisfied with their interactions is becoming more important as the Web matures. We have proposed a methodology for estimating how satisfied defined classes of customers are with a Web site. Our approach recognises that customer satisfaction is a complex issue and includes factors which are not easily measured.

We have illustrated our approach by investigating the satisfaction some typical users experience with two university Web sites. These were chosen for representative purposes only and the results do not necessarily generalise to other Web sites. Nevertheless some important observations can be made. For example, in each of our three components of customer satisfaction (complexity, time, and quality) there was a range of values for the four user classes for each site. This implies that different users of these sites will have different perceptions of how satisfied they are with their interactions with the site. We also noticed a wide range in some of the users' satisfaction components; e.g. internal students in DCU averaged 1.7 for complexity, 3.5 for time response, and 2.9 for quality on a 4-point scale. Depending on how users of this type weight these components in deciding on their overall satisfaction, they may form a quite negative impression of the Web site even though some components are satisfactory. This implies that assumptions about which components matter more to users should be checked against actual users, in order that the Web site is not optimised for some satisfaction component that users rate as less important. Overall, we have shown that our methodology is feasible and does distinguish satisfaction levels between different types of users.

The next step is to investigate whether certain "generic" categories of users can be defined, and/or whether they care about "generic" Web site parameters (e.g. it seems likely that download time will always be a factor in user satisfaction). Given a specific Web site, we will explore methods for mapping these generic user types and satisfaction parameters into the site's content. If an analysis of the resulting

satisfaction measures shows that there is a disparity in the satisfaction of different user types, we will study how the Web site designer or administrator should take this into account, and whether their reaction can be determined dynamically while the user is interacting with the site.

References

1. Nakamura et al, 'ENMA: the WWW Server Performance Measurement System via Packet Monitoring', *INET99*.
2. Cottrell et al, 'Tutorial on Internet Monitoring and PingER at SLAC' available from <http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html>
3. Kalidindi and Zekauskas, 'Surveyor: An Infrastructure for Internet Performance Measurements', *INET99*.
4. Hava and Murphy, 'Performance Measurement of World Wide Web Servers' *Proc. of 16th UK Teletraffic Symposium*, May 2000.
5. http://www.ecai.ie/usability_online.htm
6. Reid, T.R., 'WAP, Europe's Wireless Dud?' *Washington Post*, September 14, 2000.
7. <http://www.eeng.dcu.ie>
8. <http://www.fiu.edu>
9. Web Performance Incorporated, <http://www.Webperfcenter.com>

Appendix 1: Tables

Table 1: Customer Satisfaction results for www.eeng.dcu.ie

Customer Class	Trace	Parameter Satisfaction Measure			Path Weight
		Complexity	Time	Quality	
External Students	T1	4	4	4	0.5
	T2	4	4	4	0.3
	T3	2	3	3	0.2
	Avg	3.6	3.8	3.8	
Internal Students	T4	3	3	4	0.1
	T5	2	3	3	0.3
	T6	2	3	0	0.1
	T7	1	4	3	0.4
	T8	2	4	4	0.1
	Avg	1.7	3.5	2.9	
Internal Staff	T9	3	3	3	0.1
	T10	3	3	3	0.3
	T11	3	4	4	0.4
	T12	3	4	2	0.2
	Avg	3.0	3.6	3.2	
External Staff	T13	4	4	3	0.2
	T14	4	3	3	0.7
	T15	2	3	2	0.1
	Avg	3.8	3.2	2.9	

Table 2: Customer Satisfaction results for www.fiu.edu

		Parameter Satisfaction Measure			
Customer Class	Trace	Complexity	Time	Quality	Path Weight
External Staff	T1	4	3	4	0.3
	T2	3	2	2	0.4
	T3	3	3	2	0.3
	Avg	3.3	2.6	2.6	
Internal Students	T4	2	2	4	0.3
	T5	4	3	4	0.1
	T6	3	2	3	0.2
	T7	2	3	3	0.3
	T8	3	3	4	0.1
	Avg	2.5	2.5	3.5	
External Students	T9	4	3	3	0.3
	T10	3	2	3	0.4
	T11	3	2	3	0.2
	T12	3	2	3	0.1
	Avg	3.3	2.3	3.0	
Internal Staff	T13	3	3	4	0.3
	T14	4	2	4	0.3
	T15	2	3	3	0.4
	Avg	2.9	2.7	3.6	

XML Smartcards

Pascal Urien¹, Hayder Saleh¹, and Adel Tizraoui¹

Bull CP8 R&D - 68 routes de Versailles - BP 45
78431 Louveciennes CEDEX – France
{Pascal.Urien, Hayder.Saleh, Adel.Tizraoui}@Bull.net

Abstract. This paper describes how XML documents can interact with internet smart cards. Such a card works as an internet node, including a web server. Because XML is made up of entities transported by HTTP protocol, it is therefore possible to import XML entities from smart card. We describe an original process for strong user authentication, which illustrates how internet card can work with XML documents and improve security.

1 Introduction

Smartcards are generally recognized as the best device for secure computing and data storage. But until now no real efforts had been done to integrate smart cards in internet technologies. Because internet applications are based on client/server paradigm, we have developed a technology which transforms a smart card in an ordinary internet node, supporting HTTP protocol. An internet smart card can be seen as a personal web server, which can interact with XML documents, thanks to entities identified by URLs, transported by HTTP protocol and located in smart cards. Many browsers are today supporting part of XML specifications, in this paper we demonstrate that it is possible to incorporate smart card in XML documents and thus to improve security, but we emphasize that a standardization is required for smart card resources identification.

1.1 What Is a Smart Card

A smart card [2] is a portable, tamper-resistant device [3],[4],[5]; it offers safe information storage and secure processing. It contains a microprocessor (CPU), RAM, ROM, and EEPROM, the all embedded in a single tamper protected chip (whose area is about 25 mm²). It communicates with the outside world through a serial link associated to a single I/O pin, hence, only a half-duplex protocol is supported (at a baud rate from 9600 to 105900).

Although a smart card may be often physically figured out as small scaled microcomputer system (with processing unit, bus, and memory), it can not be considered as a true computer. The lack of I/O resources like keyboard, screen, etc. makes it always dependent on another computer (a terminal incorporating a card reader) which offers these resources.

ISO 7816 standards [1] define communication protocols between terminals and smart card. Embedded applications communicate with the outside world by means of Application Protocol Data Units (APDUs), which are exchanged between card and reader through a serial link according to command/response paradigm. Messages (.command) are sent from terminal to smart card, which in turn delivers a response (.response) ending by a two bytes status word.

A command APDU usually contains five bytes, CLA INS P1 P2 P3. The CLAss and INStruction bytes indicate the operation type (reading or writing for example). The two following bytes P1 & P2 provide further operation parameters (like an address), last byte P3 specifies the length of additional data bytes. The response message contains optional data bytes and ends by two status bytes SW1 & SW2. Status value "90 00" indicates an error-free operation.

1.2 Smartcards Benefits

Smart Cards are generally recognized as the most secure computers, they basically offer two kinds of features:

1. Secure data storage, files are stored and protected by various authentication methods (like pin code, mutual authentication using DES key...).
2. Secure processing , embedded cryptographic algorithms or software are executed inside this tamper resistant device.

A combined use of these two functions made it suitable for electronic signing, and authentication purposes.

2 Internet Smartcards

2.1 Goals

An internet card [10],[11] works as an internet node, and runs client or server applications defined by RFCs (like RFC 2068, HTTP 1.1 [9]). This innovative concept has been implemented in javacards, and works with existing web services; critical parameters are code byte size (around 7 kilo bytes), and data throughput of which measured value is around 300 bytes/second.

Internet card shares the TCP/IP stack of its associated terminal, from a logical point of view it acts as an internet node which uses the terminal IP address and its internet access. This sharing is achieved through a new protocol (Smart Transfer Protocol - SmartTP) which looks like a TCP [8] light protocol and connects autonomous software entities (named smart agent) located in both, terminal and card. On the terminal side, special agents (network agents) have access to network resources. On the card side, agents run internet applications (HTTP ...), and reach internet thanks to data exchange with network agents.

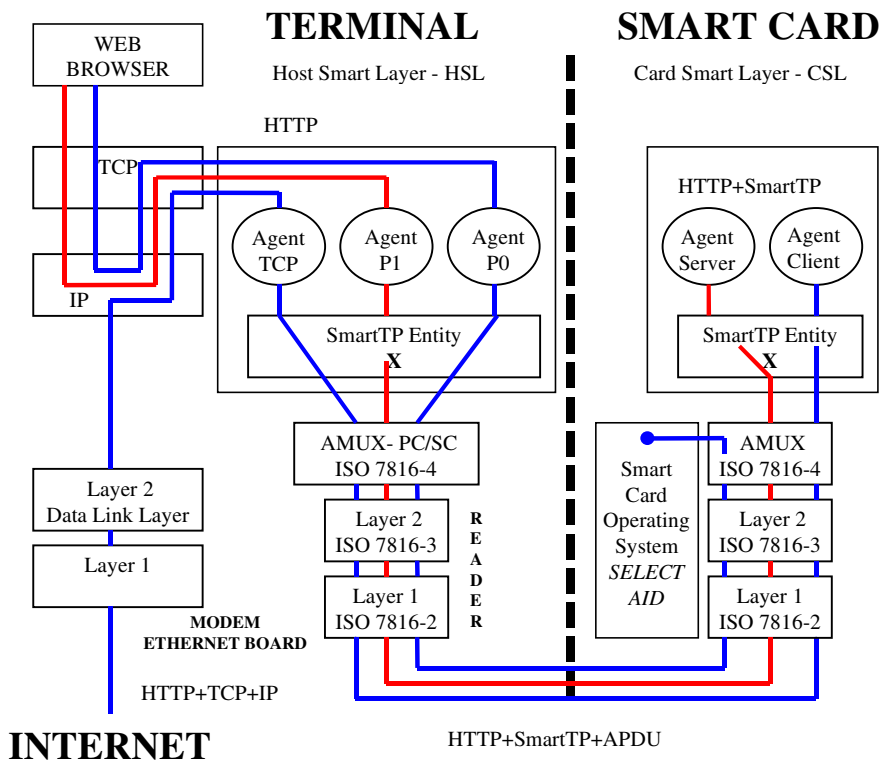


Fig.1. Internet smartcard architecture.

2.2 Architecture

Our internet smartcard architecture is illustrated in figure 1. We have defined a new communication stack, which uses two symmetric layers (Smart Layer), one located in the host system (HSL - Host Smart Layer) and the other in smart card (CSL - Card Smart Layer). HSL has access to network libraries, and to card reader APIs. It allows network packets transfer from/to the card. It establishes a logical path between existing host applications, such as web browser or electronic mail, and a smart card. CSL works with network by means of information exchanged with HSL. A smart layer is divided in two parts:

1. Smart Transfer Protocol entity (SmartTP).
2. Smart Agents.

A smart agent is an Autonomous Software Entity. It can be realized by a DLL (Dynamic Link Library) in a PC or a cardlet in a Smart Card. It's identified by a reference (a 16 bit number) which can be either constant (a well known value) or ephemeral.

In the host side agents are plugged to the network resources, they provide an internet access to agents located in a smartcard.

Agents exchange information through packets called SmartTP PDU (SmartTP protocol data unit). SmartTP entity is a logical switch, and is in charge of routing incoming or outgoing PDUs to/from agents.

The communication stack (figure 1) used by a network card and its associated terminal is the following,

1. OSI layer 1 and 2 (ISO7498 [7]), supporting ISO 7816-3 transmission protocols.
2. AMUX layer (Apidu multiplexer, using either PC/SC [6] or ISO 7816 [1] services), which routes APDUs to/from SmartTP entity.
3. SmartTP entity, which switches SmartTP pdu towards agents.
4. Agents, which process application data, and exchange SmartTP pdu.

2.3 Basic Applications

Web server

A web server is an internet protocol specified by an RFC standard (HTTP 1.1 [9]). Its implementation in a card means that HTTP data, which are carried through the web by TCP/IP packets, are exchanged between card and terminal by means of smartTP PDUs. From the application point of view an HTTP session is opened between the client (a browser) and a web server located in the card.

An URL, *http://127.0.0.1:8080*, where *127.0.0.1* is the terminal IP loop back address and *8080* the network agent TCP port, gives access to card index (an HTML file), which includes hyperlink towards internal or external resources. Embedded card resources, like cryptographic entities (cipher algorithm, digital signature, authentication procedure), multimedia objets (html page, image, sound...), software's (java applet...) are identified by URLs.

Trusted proxy

A proxy is a powerful and useful entity in the world of the TCP/IP technology. It includes a static TCP server and a TCP client, which is created dynamically upon each new incoming connection to the server. Client establishes a connection to a node, either a pre-defined one, or which is deduced from information's received over the server connection. A proxy forwards application data (carried by TCP/IP packets) from a TCP connection to another. A trusted proxy (embedded in a smart card) may be used for security purposes (SSL proxy, firewall), or to perform protocol translation.

2.4 Working with Internet Smartcard

Usually a smart card includes several embedded applications which are identified by a 16 bytes number named application identifier (AID). Therefore a specific APDU (*SELECT AID*) is required to activate a given application.

We have defined a three levels architecture (figure 1) in order to work with an internet smart card.

1. First level is a network agent (agent p0) associated with a TCP port p0 (for example p0=8082), which implements a web server and which is used to manage a

- smart card. Typically it is possible to select by a particular URL (like `http://ip:p0/?write`, see figure 3) an application located in the card.
2. Second level is a network agent (agent p1) associated with a TCP port p1 (for example p1=8080) and which is used to route HTTP request message (`http://ip:p1`) towards a smartcard agent implementing a web server.
 3. Third level is a network agent (TCP agent) , which is used by smart card to establish TCP (client) connection with a remote internet server.

3 Interactions between XML Documents and Internet Card

3.1 XML Document and Smartcard Interactions

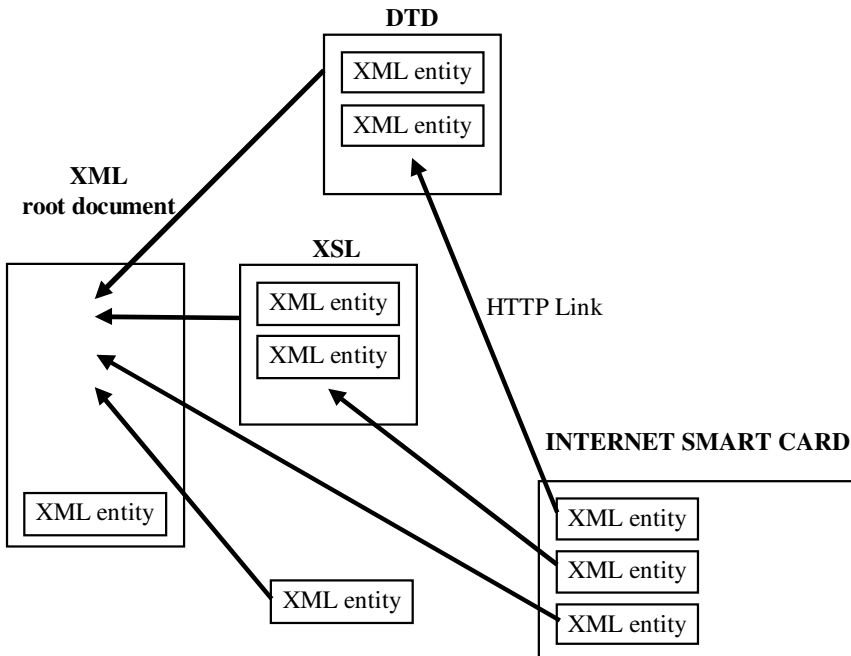


Fig.2. XML documents and internet smartcards

XML documents [12] are made up of storage units called entities, which if necessary, are transported by HTTP protocol; we propose to import some of them from an internet smart card (figure 2). From a logical point of view an XML document is a tree of one or more elements, the boundaries of which are delimited by tags.

Each element has a content which can be either an element or an another XML object (like entity or character data).

A data type declaration (DTD), contains or points (by means of an URL) to markup declarations that provide a grammar for a class of document. A DTD defines

the tree structures, which are allowed by the XML document issuer. DTD can be made up of several entities, some of them may be embedded in an internet smart card.

Unlike html page, XML documents can't be directly displayed, an extensible style sheet language file (XSL [13]) is generally needed to build an HTML (or WML) page, which is a human representation of some element contents. This HTML page may include software components, like script or applet, which will be loaded and executed at run time by a web browser.

For example a javascript will force a redirection (deduced from XML elements contents) to an other web site (by invoking the location.href method), or an applet will process data imported from the original XML document.

Interactions between an XML document and an internet smart card occurs in three steps (figure 2),

1. First a browser downloads a root XML document which includes pointers to several physical XML entities, like XML document fragments, DTD and XSL. Entities are identified by URLs, some of them are located in one or several smart cards. The complete XML document is linked and then checked by a parser according to definitions found in its associated DTD.
2. Second an html page is build by the XSL processor, this page can include scripts or applets which will be invoked at run time with calling parameters deduced from elements contents.
3. Third a browser loads the produced HTML page, which includes software components like script or applet process.

3.2 Internet Smart Card Detection

A basic request, from a server point of view is to determine if a smart card is available on a given terminal or not. A possible solution to this problem is to use a technique that we call *card bug*. A card bug is identified by an URL which points to an image file (of which size is typically one pixel, for example a white pixel). An HTML page is able to detect (as shown in figure 3) an image downloading, and according to this event to dynamically select an XML (or HTML) document.

Card bug (figure 3) is used to detect and select a particular application embedded in smart card. As an example the URL

`http://127.0.0.1:8082/?write=00A40400054A54455354`

(sent to the p0 agent) selects (Select APDU = 00 A4 04 00, length = 05) an JTEST application (AID = 'J' 'T' 'E' 'S' 'T' = 4A 54 45 53 54).

A *card bug* embedded in a smart card (like `http://127.0.0.1:8080/key1.gif`) is used to indicate the availability of a DES key whose name is key1.

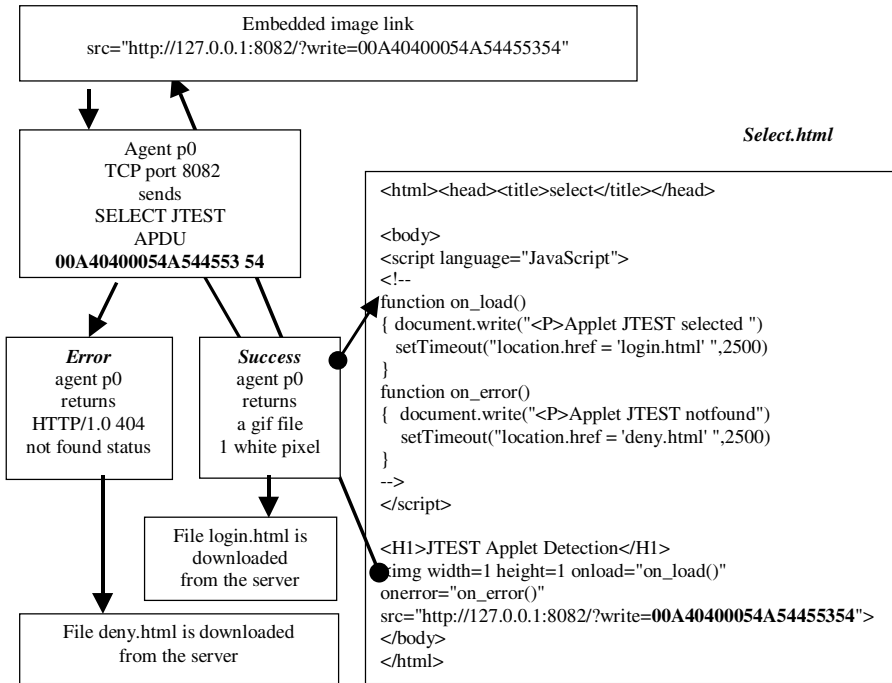


Fig.3. Card bug concept.

3.3 Example of Internet Card Resources

Table 1. Internet smartcard resources

File name	Meaning	Format
/	Smartcard index	HTML page
/name.txt	Bearer name – Pascal Urien	XML entity
/key1.gif	Card bug	GIF file
/Key1=69DA379EF99580A8F	DES encryption of a 8 bytes block	XML entity
/Key1+=69DA379EF99580A84	DES ⁻¹ encryption of a 8 bytes block	XML entity

We have designed an internet javacard (of which code byte size is around 7 kilo bytes) which includes a web server, and services identified by URLs. Table 1 shows the embedded resources list.

Our smart card includes an index page, which contains information about its content, the bearer name (name.txt), a card bug (key1.gif) which indicates the availability of a DES key whose name is key1, and a method to compute DES (and DES⁻¹) algorithm according to the key1 value.

3.4 Strong Authentication

A strong authentication process is illustrated in figure 4. An html page of which name is *select.html* is downloaded by a browser. In this page a card bug

```

```

tests the presence in the smart card of a DES key named key1. If this key has been detected the browser loads an XML document, login.xml.

This document includes an entity (the content of *name* element) whose identifier is the bearer name (&name; - http://127.0.0.1:8080/name.txt) and an other (the content of *response* element) which requests the card to cipher a random number (the content of *challenge* element, 1234) with the card DES key1 algorithm (http://127.0.0.1:8080/Key1=+1234). Once all parts of the XML document have been collected, an XSL processor builds an HTML page which is displayed by the browser. This page shows the bearer name (Pascal.Urien), the random number (1234) and the DES computation (response) of this number (DES(1234) = 5702C18C3A056058). These data are gathered by a script, which forces the browser to download a new html page whose name (name.random.response) is deduced from XML entities contents (in our example the requested page is, Pascal.Urien.1234.5702C18C3A056058).

4 Conclusion

We have demonstrated that internet smart card can interact with XML documents. We think that this innovative concept could improve security over the internet and could be used to extract private information from XML document. Obviously this technology required standardization efforts, first smart card URL formats need to be formerly defined (what are the values of the associated TCP port ?), and second card embedded resources should be identified by well known DTD coupled with card bug.

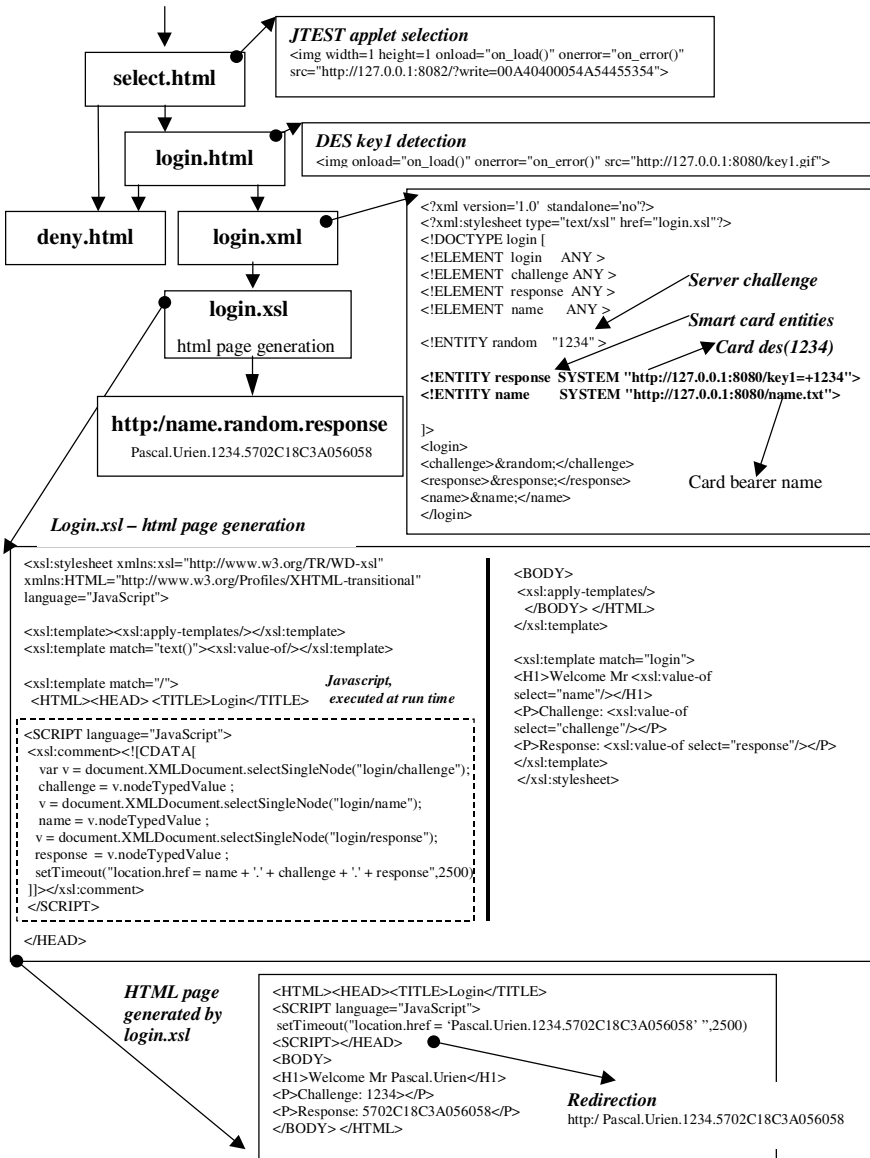


Fig.5.Strong authentication process.

References

- [1] International Organization for Standardization (ISO) "Identification cards - Integrated circuit(s) card with contact" ISO/IEC 7816.
- [2] Louis Claude Guillou, Michel Ugon, and Jean-Jacques Quisquater, "The Smart Card: A Standardized Security Device Dedicated to Public Cryptology," Contemporary Cryptology. The Science of Information Integrity, ed. Gustavus J. Simmons, IEEE Press 1992, pp. 561-613
- [3] R. Anderson, M. Kuhn " Tamper Resistance - a Cautionary Note", USENIX Workshop on Electronic Commerce Proceeding, November 1996, pp 1-11.
- [4] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", Advances in Cryptology: Proceedings of EUROCRYPT'97, Springer-Verlag, August 1997, pp 513-525.
- [5] Paul Kocher, Joshua Jaffe, and Benjamin Jun "Differential Power Analysis" Proceedings of CRYPTO'99, Springer-Verlag, August 1999, pp 388-397.
- [6] "Interoperability Specification for ICCs and Personal Computer Systems", PC/SC, © 1996 CP8 Transac, HP, Microsoft, Schlumberger, Siemens Nixdorf.
- [7] International Organization for Standardization - (ISO) "Information Processing Systems - Open Systems Interconnection - Basic Reference Model" ISO 7498.
- [8] Postel, J., "Transmission Control Protocol", RFC 793, September 1981.
- [9] T Berners Lee & All "Hypertext Transfer Protocol - HTTP/1.1" - RFC 2068 - January 1997.
- [10] Pascal Urien " Internet Card, a smart card as a true Internet node", Computer Communications, volume 23, issue 17pp 1655-1666- October 2000.
- [11] Pascal Urien, Hayder Saleh, Adel Tizraoui "Internet Card, a smart card for internet", Protocols for Multimedia Systems (PROMS) Cracow Poland, October 22-25 2000.
- [12] Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000.
- [13] Extensible Stylesheet Language (XSL) Version 1.0, W3C Candidate Recommendation 21 November 2000.

The Influence of Web Page Images on the Performance of Web Servers

Cristina Hava Muntean, Jennifer McManis, and John Murphy

Performance Engineering Laboratory, School of Electronic Engineering,
Dublin City University, Glasnevin, Dublin 9, Ireland
{havac, mcmanisj, murphyj}@eeng.dcu.ie

Abstract. In recent years World Wide Web traffic has shown phenomenal growth. The main causes are the continuing increase in the number of people navigating the Internet and the creation of millions of new Web sites. In addition, the structure of Web pages has become more complex, including not only HTML files but also other components. This has affected both the download times of Web pages and the network bandwidth required. The goal of our research is to monitor the download times of Web pages from different Web sites, and to find out to what extent the images contained in these Web pages influence these times. We also suggest some possible ways of decreasing the bandwidth requirements and download times of complex Web pages.

1 Introduction

At the beginning of the World Wide Web in the early 1990s, most Web pages were text based, with file sizes on the order of hundreds of kilobytes. Nowadays, Web pages have become much more complex. Static and animated pictures, sounds, dynamically generated pages and multimedia components have been included, increasing the typical total size of these Web pages to megabytes. In this way Web pages have become more attractive for their clients, but also more resource-intensive to send and retrieve. The immediate effects are increased delays in accessing the documents and overloading of the network. Thus fewer users are able to access Web site information in a given time period. At the same time Web traffic has become the most common type of traffic on the Internet, and the number of Web sites continues to increase dramatically.

Most users who navigate on the Internet want to access a Web site as quickly as possible and don't have the patience to wait a long time to load overly large Web pages. In order to increase the number of the clients who access a site and to keep those who are already visiting it, the owners of commercial sites should constantly monitor the performance of their servers. One of the most critical parameters is the download time, which gives a good indication of the waiting times for potential clients.

Because a Web page consists of not only one HTML file, but a collection of many file types (e.g. HTML, images, JavaScript, Active Server Pages (ASP), cascading style sheets (CSS), MacroMedia's Shockwave), the page download time is the time to download all the Web page's components. While it is possible that the user might care

about the download times of individual components, we believe the total download time is in general important.

We have monitored the download times for different commercial sites, and analysed their composition with respect to file size and type. Our experimental results show that images represent the biggest percentage of Web page size, and hence account for a considerable proportion of the download time for the page. We observed that some sites use dynamic HTML pages generated by JavaScript files or tags and Active Server Pages to minimize download times. These types of files are run on the client machine and can produce the same effect as a static HTML file with a lot of images inside, but are relatively small in size.

The main aim of our research is to determine which components make the largest contribution to the total download time. We also suggest some possible solutions to decrease Web page download times and to reduce overloading of the network.

2 The Structure of the Web Pages

Web pages are composed of multiple object documents. The main document is an HTML object, which can consist of one or more HTML files. The other objects are inline images or animations and Java applets. A browser accesses all these objects from the Web server using the HTTP protocol. The number and size of the object documents embedded in the Web page influence the download time of the Web page.

Two types of the HTTP protocols are in use. The HTTP/1.0 protocol retrieves objects using a separate TCP connection for each object retrieved. Thus, multiple connections are created simultaneously between the server and the browser. As the number of components increases, more requests must be sent to the server, thereby increasing the total download time. Along with all the requests sent by other clients, these could easily overwhelm the server. The second protocol, HTTP/1.1 supports persistent connections between the server and the client. In this case a single connection is used and all the document objects are delivered sequentially. The use of the HTTP/1.1 protocol reduces the connection overhead by using a single connection for getting all the components. However, the sequential nature of the retrieval might reduce the performance improvement in the case of many components.

We analysed a number of Irish commercial Web sites, to determine the structure of their Web pages and the contribution of their components to the overall size. Specifically, we determined the structure of the main Web page (total size of the Web page, number of images, the percent of HTML and image files). The results are listed in Table 1.

For the analysed Web sites, we see that - with the exception of Web Server 6 and Web Server 11 - images represent by far the biggest component of the Web pages. Some of the pages also have a large *number* of images. Both size and number of images could affect both network and server performance, especially in peak hours when there are a lot of clients visiting the page. Apart from images, the studied Web pages included other multimedia components such as JavaScript, ASP, and MacroMedia's Shockwave files.

To find out how images influence the access time of the Web pages, we did different tests for the sites in Table 1. In order to isolate those factors relating strictly to the composition of the Web page, we also analysed the structure of different Web

pages from the same site. In this way, we could account for the influence of the network path and the performance of the server machine. These tests and their results are presented in the next section.

Table 1. Statistics about the composition of the studied Web pages

Sites	Total size (KB)	Html file size (%)	Images size (%)	Other size (%)	Number of Images
Web Server 1	368.5	1.06	98.94	0	2
Web Server 2	331.6	4.70	88.53	6.77	90
Web Server 3	136.9	8.66	73.90	17.44	26
Web Server 4	71.3	3.53	76.26	20.22	13
Web Server 5	72.0	9.30	90.70	0	8
Web Server 6	113.2	0.75	51.01	48.24	59
Web Server 7	57.1	17.27	68.53	14.11	14
Web Server 8	78.9	0.23	86.33	13.44	28
Web Server 9	117.6	7.28	92.71	0	6
Web Server 10	86.6	13.69	85.83	0.48	43
Web Server 11	51.5	13.66	33.39	52.95	88

3 Experimental Results

For our experiments we used the commercial sites presented in Table 1. These sites span a range of sizes, but all of them contain HTML, images, and other types of files. Download times are measured using a tool developed in the Performance Engineering Laboratory at Dublin City University [1, 2]. In our first experiment, we measured the effect of images on download times by comparing download times of the main HTML file for each site with that of the main page in its entirety. In our second experiment, we measured the effect the number of images had on download time under different network and server loadings. Different loadings are achieved by taking measurements throughout the working day. In our third experiment, we attempt to isolate the effect of page composition on performance from other factors of network and server loading. This is achieved by testing different Web pages from the same site.

3.1 Experiment 1: Effect of the Images on Download Times

To find out how much the images of a page influence the download time, we made a comparison between the download time of the main page and the download time of the main HTML file. As an example, the load times of the main Web page from the Web Server 5, with and without the images, are shown in Figure 1.

We observe that the time necessary to download the main page and all its associated files is approximately four times as large as the time to download just the main HTML file. This increase is due to the eight images that are part of the page. These eight images represent 90.7% of the size of the Web page. In addition to taking a long time to download, these images are responsible for a significant increase in Web traffic.

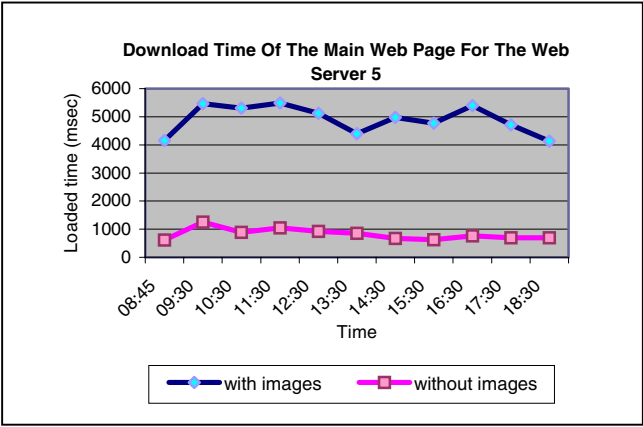


Fig. 1. Comparison of the download time of the main page from Web Server 5 with and without all its components

3.2 Experiment 2: Sensitivity of Performance to Web Page Composition

In our second experiment we demonstrate that Web pages with a large number of images are more sensitive to network and server loading than those with fewer images. We compare measurements for four different pages having a large difference in the number of images. Two of the main pages (Server 5 and Server 9) have less than ten images, while the other two pages (Server 11 and Server 2) have around ninety images each. We periodically monitored the download time for the pages during a weekday, between 8:45 am and 6:30 pm. The download time at 8:45 am is taken to be a baseline measurement and a growth factor is measured as the ratio of the current download time to the download time at 8:45 am. Our results are summarised in Figure 2.

As can be seen, the pages with a large number of images had a much larger growth than the pages with a small number of images. This indicates that a large number of images can seriously affect Web server performance.

3.3 Experiment 3: Effect of Number and Size of Images on Download Time

In Experiment 2, many factors might have influenced the download time including not only the Web page composition, but also the server performance, the network traffic and the distance from the client to the Web server (although all the Web servers are located in Ireland). In order to isolate the effect of page composition on the download time, we compare different pages from the same site. Pages are chosen with different numbers and/or sizes of images. First we look at pages that have varying numbers of images, but are all of similar size and with similar percentage of the size being accounted for by images. Second we look at pages where both number and size of images vary. In both cases the download times are measured for various server loadings, and the relative degradation of performance is obtained.

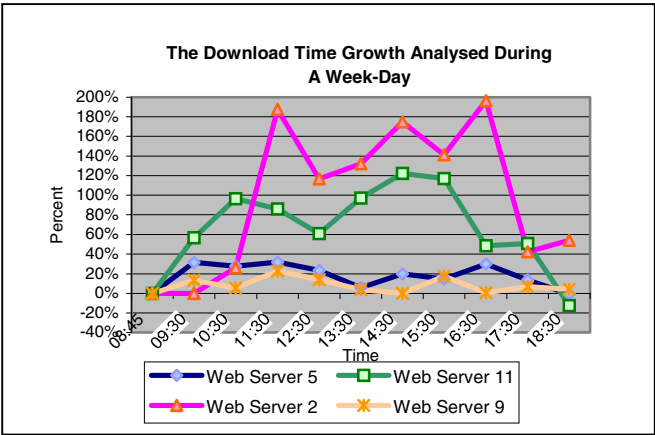


Fig. 2. Download time growth for different Web pages during a day

From Web Server 2 we chose three different Web pages with different numbers of images (between sixty-eight and ninety), but with similar image sizes as a percentage of the total size of the Web page. Because most of the problems of the Web servers' performance appear during the peak-hours period, we analysed the response time of the server for that period. Server loads were generated by making parallel requests for the page. Measurements are taken for 1, 10, 30 and 100 parallel requests. These measurements are summarised in Table 2. Growth is defined as the ratio of the current download time to the download time for a single client request.

Table 2. The average download times during the peak hours of different Web pages from Web Server 2 with a variable number of simultaneous accesses

Web Page Number	Page Size (KB)	Num ber of Imag es	Img. Size (%)	Numbe r of Parallel Clients	Averag e Download Time (sec)	Growt h
Page 1	292.5	90	87.51	1	14.88	1.00
Page 1	292.5	90	87.51	10	18.85	1.27
Page 1	292.5	90	87.51	30	25.75	1.73
Page 1	292.5	90	87.51	100	54.39	3.66
Page 2	231.8	75	88.89	1	11.34	1.00
Page 2	231.8	75	88.89	10	12.86	1.13
Page 2	231.8	75	88.89	30	18.89	1.67
Page 2	231.8	75	88.89	100	37.27	3.29
Page 3	209.4	68	89.65	1	12.25	1.00
Page 3	209.4	68	89.65	10	13.50	1.10
Page 3	209.4	68	89.65	30	18.80	1.53
Page 3	209.4	68	89.65	100	39.04	3.19

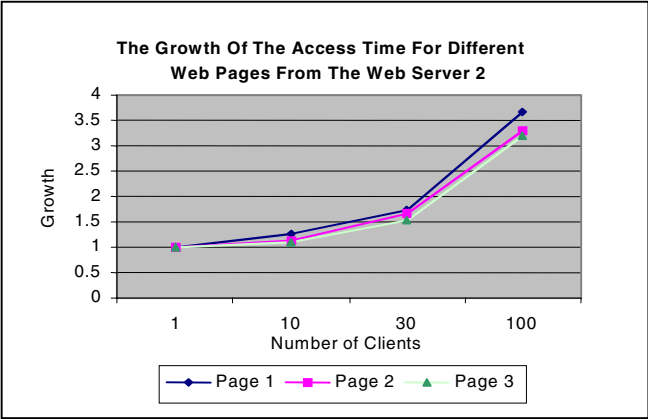


Fig. 3. Access time growth during peak-hours for different pages from Web Server 2

When the number of clients who access the same Web page in parallel increases, the growth factor of the download time is bigger. Thus, more and more requests for the components of the Web page are sent to the server overloading it. Comparing Web Page 3 with Web Page 1 there is a significant difference of the download time growth when there are 100 clients in parallel. The growth factor for the three pages is presented in Figure 3.

We see that Page 1 has the worst performance and it has the largest number of images. Page 1 is also slightly larger than Page 2 and Page 3, with a larger image size.

Table 3. The access time of Web pages with various numbers of images from Web Server 8

Web Page Number	Page Size (KB)	Num ber of Images	Img. Size (%)	Numbe r of Parallel Clients	Average Download Time(sec)	Grow th
Page 1	78.9	27	86.29	1	7.90	1.00
Page 1	78.9	27	86.29	10	8.83	1.12
Page 1	78.9	27	86.29	30	19.65	2.49
Page 1	78.9	27	86.29	100	42.19	5.34
Page 2	43.5	39	54.37	1	2.78	1.00
Page 2	43.5	39	54.37	10	3.98	1.43
Page 2	43.5	39	54.37	30	7.30	2.62
Page 2	43.5	39	54.37	100	19.09	6.87
Page 3	36.9	17	53.20	1	2.23	1.00
Page 3	36.9	17	53.20	10	2.55	1.14
Page 3	36.9	17	53.20	30	4.01	1.80
Page 3	36.9	17	53.20	100	11.62	5.21

In order to study the effect of the size of the images versus the number of images on download time, a similar analysis was done for three Web pages from Web Server 8. For these pages, both the size of the images and the number of images varies. The composition of the Web pages is summarised in Table 3.

A comparison between the download time for Page 2 and Page 3 shows the influence of the number of the images. Although Page 2 and Page 3 are similar in size and size of images, at 100 requests Page 2 takes nearly twice as long to download as Page 3. The growth factors for the three pages are plotted in Figure 4.

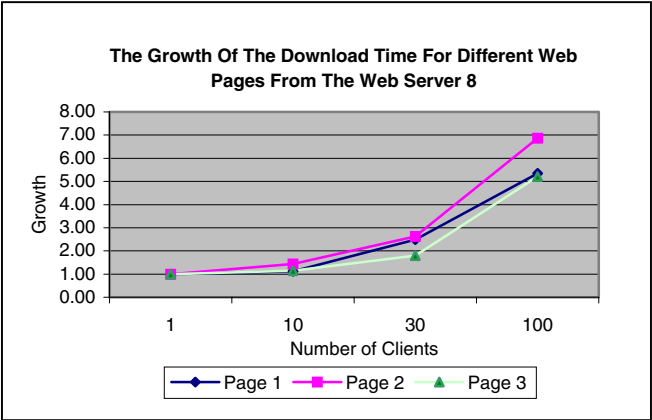


Fig. 4. Growth factors for different Web pages for Web Server 8

A comparison of Page 1 and Page 2 indicates that the number of images has a greater influence on performance sensitivity than the size of images. Page 2 is smaller, but has a greater number of images than Page 1. The growth factors for Page 2 are consistently larger than those for Page 1.

4 Conclusions and Future Work

The results of this study lead to a number of interesting observations about some factors that could influence Web Server performance. These factors include: the number of images, the total size of the images, a large number of clients accessing the Web server simultaneously, and the period of time (peak/off-peak hours) when the requests are made. The work reported here suggests that the number of images has a disproportionate effect on server performance, particularly when the server is heavily loaded. In order to ascertain if our assumption regarding loading patterns is correct, it will be necessary to either measure or control the loading of the Web pages.

Experimental results suggest that images do have a great influence on download time. This indicates that designers of Web pages need to find a compromise between the look of a page (with lots of attractive pictures) and the performance seen by clients of the page (for which download time is a reasonable measure). Many *static*

solutions exist to improve download time: for example, a faster Internet connection, a better-performing server, and smaller Web page sizes. A significant amount of effort has gone into minimizing image sizes and bandwidth requirements. A lot of research on compression algorithms has been done suggesting that one may reduce the size of an image file, keeping a good image quality [3]. Also UC Berkeley's Transend [4], Intel's QuickWeb [5] and Spectrum's FastLane [6] systems tried to improve the access to slow links reducing image size via lossy compression using Web proxies which transform the images in ones with resolution and color reduction. Gilbert and Brodersen [7] proposed a methodology to improve Web access using a new technique called global progressive interactive Web delivery, which entails applying progressive coding to the document transmission process in its entirety.

Another solution is to use DHTML animations created with JavaScript, Macromedia's Shockwave/Flash, or Microsoft's DirectAnimation instead of image files (currently, most of the images on the Web are GIF or JPEG [8]). The effect of these files is more spectacular and their use decreases the number of connections created between the browser and Web server, thus reducing bandwidth requirements. Other possible solutions to improve the download time are presented in [9] where ways are suggested to reduce the number of bits each page needs and to make the JavaScript code faster.

We suggest that a class of *dynamic* solutions should also be considered. For example, the Web server could monitor its download times and reduce the amount of information sent during peak times. Transmitting only some of the embedded files will reduce the Web page content quality. In this case a compromise between the quality of the Web page and the performance of the server has to be made. It may also be possible for the client to monitor the speed of the download and control how much information they want to receive. In this way the client's perception of the Web page would take into account the page's size and composition, and how these affect the expected waiting time.

References

1. Cristina Hava, Liam Murphy, "Performance measurement of WWW Servers", *16th IEE UK Teletraffic Symposium*, Harlow, United Kingdom, May 2000.
2. Cristina Hava, Stefan Holban, Liam Murphy, John Murphy, "Initial tool for monitoring the performance of Web Sites", *CONTI'2000*, Timisoara, Romania, October 2000.
3. Uwe Rauschenbach, "Compression of Palettized Images with Progressive Coding of the Color Information", *SPIE2000*, Perth, Australia, June 2000.
4. Fox Armando, Steve Gribble, Yatin Chawathe and Eric Brewer, "The Transend Service", <http://transend.cs.berkeley.edu/about>
5. Intel Corporation, "QuickWeb homepage", <http://www.intel.com/quickWeb>
6. Spectrum Information Technologies homepage, <http://www.spectruminfo.com>
7. J Gilbert and R. Brodersen, "Globally progressive interactive Web delivery", IEEE Infocom'99 Conference, New York, March 1999.
8. Adrian Vanzyl, "Increasing Web Bandwidth through Image Compression: An overview of GIF, JPEG and Fractal Compression Techniques", *AusWeb95*, <http://www.scu.edu.au/sponsored/ausWeb/ausWeb95>
9. Michael Wallent, "Frequent Flyers: Boosting Performance on DHTML Pages", *MSDN Online Voices*, <http://msdn.microsoft.com/workshop/>

Network Resilience in Multilayer Networks: A Critical Review and Open Issues

F. Touvet and D. Harle

Strathclyde University, Glasgow G1 1EX, Scotland
f.touvet, d.harle@eee.strath.ac.uk
44-(0)141-548-2090

Abstract. Current networks integrate multiple transport technologies to exploit their relative advantages and different functionalities, and these are multilayer networks combining IP/ATM/SDH/WDM technologies. To provide end to end survivability, different restoration mechanisms at different layers must be combined efficiently in order to achieve fast restoration, maximum availability and minimum restoration cost. This paper describes the key issues associated with the implementation of restoration in a multilayer network. Moreover, the “best” strategies proposed so far are presented and the current state of standardisation is discussed. Finally, current unresolved issues and open problems are highlighted.

Keywords: survivability, multilayer network, resilience issues, restoration strategies

1 Introduction

With the deployment of networks combining up to four different technologies (IP, ATM, SDH/SONET and WDM) for the network layer, the design of end to end survivability in multilayer networks has become a major topic of interest in itself. A standardised infrastructure for multiple network layers is presented in [17,20]. New issues associated with survivable multilayer networks arise; these issues are related to the layering and the fact that each technology provides its own restoration mechanism. The layering encompasses the overlap of functionalities, and adds complexity of interworking. Issues of paramount importance include the coordination between the various restoration mechanisms at different layers and the allocation of spare resources. Also, issues regarding the timing and location of the restoration process need to be addressed.

A number of projects have been set up, as the issue of survivability in multilayer networks attracted significant interest. European projects like ACTS PANEL, MISA or RACE II IMMUNE [25] led to a better understanding of the issues of restoration in multilayer networks, and the design of a general framework for designing end to end survivable networks. A survivability framework can also be found in [25]. These projects also contributed to the process of standardisation.

The remainder of the paper compares the results of different projects and concludes with a recommendation of the “best” strategy for certain purposes. In spite of the encouraging results obtained so far, it is noted that most of the strategies developed were only proposals demonstrated and are not yet implemented in real life networks. In other words, the state of standardisation is still in its infancy. Some standards that have been initiated are highlighted.

While the major issues are well known, some problems still remain unresolved and preclude the restoration design from being cost effective and satisfactory. Interesting unresolved issues are expressed in this paper.

2 Comparisons of Strategies

End to end survivability in multilayer networks implies consideration of restoration mechanism, i.e the restoration algorithms or the deployment of 1+1, 1:1 or 1:N protection¹. Survivability is designed to enable the recovery of traffic flows affected by network failures.

Resources are added to allow for the traffic to be carried over spare resources until the failed equipment is repaired. Allocating spare capacity is complex as such capacity is needed at each layer and becomes even more complex when sharing capacity between layers is considered. The cost of spare capacity should be kept to a minimum while simultaneously ensuring maximum survivability. A further cost of restoration is the exchange of restoration messages that require extra overhead management.

The allocation of spare capacity is closely related to the strategy of coordination of each recovery mechanisms, as well as the failure scenarios considered. The coordination of recovery mechanisms, also known as escalation [28], is essential to provide complete restoration at minimum cost. Additionally, the coordination seeks to achieve minimal restoration time. Thus, the role of the Telecommunications Management Network (TMN) is considered. Finally, any strategy needs to take into account different granularities which, potentially have different values at each layer. Granularity characteristics involve issues such as the time scale for the restoration that varies between milliseconds and minutes(temporal granularity), the rerouting can be processed at the packet level or at the wavelength level(bandwidth granularity), and aggregated traffic classes or individual traffic classes can be considered(QoS granularity).

A number of strategies have been proposed. First of all, the issues of where and when to trigger the recovery mechanisms are considered. It is stressed at this point that before considering restoration in a multilayer network, the most suitable restoration strategy must be selected at each layer. For ATM and SDH technologies, [22,26] report an analysis for the choice of the recovery strategy. At the optical layer, different protection strategies are referenced in [23]. Lastly,

¹ dedicated resources are allocated. For 1+1, traffic is sent in both the working and backup paths simultaneously. For 1:1 and 1:N, the traffic is sent in the backup path after a failure in the working path has been detected. 1:1 corresponds to 1 backup link for 1 working link, whereas 1:N corresponds to 1 backup link for N working links

other references can be found in [3]. Restoration mechanisms are required at each layer, e.g to cope with cases of failures of cross-connects at each layer. To avoid deadlock situations and resource competition, recovery mechanisms at each layer need to be coordinated [34]. So far, two types of strategy have been considered to achieve end to end survivability: the “lowest layer recovery strategy” and the “highest layer recovery strategy” [7,11,15,16]. The “lowest layer recovery strategy” starts the restoration at the closest layer to the failure location. The “highest layer recovery strategy”, instead, starts the restoration at the closest layer to the origin of the traffic.

In the ACTS PANEL Project, both strategies have been compared and the “lowest layer recovery strategy” appeared to be the most efficient in both achieving faster restoration and lower cost resources.

However, this strategy is not the best suited strategy for providing survivability in multilayer networks. Indeed, this strategy is more complex to implement, as coordination between the recovery mechanisms within each layer is required [2,5]. Moreover, signaling messages for the coordination must be defined [25]. After the detection of defects, recovery mechanisms must be triggered by the intervention of TMN to prevent the undesirable triggering of mechanisms at other layers before the server has the opportunity to complete the restoration.

The coordination between the recovery mechanisms is realised by delaying the activation of the recovery mechanisms at higher layers. Two means have been employed: the hold-off time and the recovery token. Demonstrations in the labs led to the conclusion that the use of a hold-off time enables a faster restoration. Therefore, a satisfactory strategy to tackle the problem of end to end survivability would be to implement a hold-off timer which delays the activation of the higher layer recovery. Thus a maximum amount of traffic is restored by the lowest layer. The hold-off time is set up to prevent deadlock situations, and can be modified in the TMN within the Performance management area [25].

The PANEL project [11] considered an integrated management system that enables the coordination of the overlay layers. An SDL tool from the University of Munich (SELANE) was used, among other things, to model the integrated management system. An Integrated Network Management (INM) system can be modeled, to act as an end-to-end management control of the layered network. The management system referred to the Q3nn ATM and SDH management interfaces developed during the project ACTS MISA.

Within BT, a protocol was developed to achieve end-to-end survivability in hybrid ATM/SDH networks.

In their work, Veitch et al. [31,32,33] dealt with the restoration of traffic in an ATM/SDH network, considering an approach similar to the “highest layer recovery strategy” developed by the PANEL Group, and corresponds to a groomed multilayer policy. Restoration processes were decoupled and processed concurrently. Only physical failures were considered, as they are the most likely to occur in practice. The results of the developed protocol proved full resilience against physical failures. However, the time needed for restoration was not evaluated.

In other experiences, the cost in terms of spare capacity required was compared for different strategies of restoration. In [33] Veitch et al. showed that it is cheaper to restore client connections at the client layer rather than at the server layer, and this strategy becomes even cheaper when the number of client cross-connects increases. These results are in contradiction to the ones obtained in the PANEL project. However, it must be stressed that the experiments only considered single link failures, whereas also link and cross-connect failures are taken into account in PANEL. Besides, in PANEL the number of overlays of different technologies was considered small, whereas in BT, a full overlay of one technology on top to another was considered. In [33], one drawback was pointed out: restoration in the highest layer results in a higher number of connections to restore², hence the time to restore might be long, and the management of the restoration messages be complex. Therefore, to increase the granularity, a proposition of grouping the client connections was made (consisting of VPGs (Virtual Paths Groups) for ATM VPs) [20,32].

In the BT approach the spare capacity was not a major concern. Requirements of each layer were calculated separately. As the demand of spare capacity from higher layers must be carried by lower layers, the cost can be very high. To lower the cost, the PANEL Group proposed to share the spare capacity between layers. The concept is meant to be generic and is known as “common pool of capacity” [5,13,14,15]. In the PANEL project, the common pool was proposed for ATM/SDH and SDH/WDM [6] layered networks. Comparisons of costs proved that the common pool allows important savings. However, the concept considers only a single failure, a cable cut or a cross-connect failure. Moreover, the common pool applies to the “lowest layer recovery strategy”. The concept is possible when the spare capacity allocated at the client layer is considered as preemptible resource. Therefore, a protocol to perform the preemption is required. A “squenching” mechanism which informs the client layer of the unavailability of its spare resources, then enabling the preemption, is standardised for SDH rings [18].

Both BT and PANEL approaches provided frameworks to consider the survivability in multilayer networks. A guideline resulted from the PANEL project: NIG-G5 ACTS, “Towards Resilient Networks and Services”.

Moreover, the PANEL project contributed to the work of the ITU-T SG13 Q19/13 WP3/13 working on the adaptation of the SDH layer to OTN layer networks. Besides, PANEL has influenced the work of the ETSI TM3 WG13, which released the DTR/TM-3025 paper related to the hold-off time functionality in MSP and MS-SPRing. More information concerning the hold-off timer is found in the draft technical report “Enhanced Network Survivability Performance” from the Working Group T1A1.2, released in November 2000 [9].

Recently, more consideration is given for restoration in optical IP networks. Survivability considerations for such networks are provided in [27]. Moreover, restoration in optical IP networks has been described for different networking architectures in [3]. Rather than considering restoration in each independent

² due to the finer bandwidth granularity of the highest layer

networking layers (optical and IP), it is shown in [3] that integrated architectures are more cost effective in terms of required equipment. Moreover, analogous to the project PANEL where “highest layer recovery” and “lowest layer recovery” schemes were compared, [3] compared the cost effectiveness between restoration in a “service layer” architecture and a “transport layer” architecture. The drawn conclusion was that the latter is the most suitable for large IP networks, providing better restoration performance than service-layer architecture and at a efficient cost.

Clearly, significant work has already been carried out and major results been achieved. However, some issues are still unresolved. The next section discusses some of those.

3 Open Issues

Studying approaches presented in the literature, some open issues have been derived. The following list presents problems found out during the various studies, and still unresolved. Initial work has been carried out to address some of them. Clearly, this list is not exhaustive but some of the major issues are included.

The issues can be classified into three different parts.

1. Rapid detection of the fault, crucial for a restoration strategy based on a “highest layer recovery” approach.
2. Management of the restoration mechanisms for an end-to-end survivability based on a re-routing approach.
3. Minimisation of the cost of spare capacity.

The problem of detection of failures in IP networks is still an issue, as with current protocols the time it takes to detect a failure at the IP layer is still typically measured in tens of seconds [1,3,24,35]. Such a long time-to-detect is harmful to end-to-end restoration. First of all, the large time to detect might affect a large amount of traffic, hence dropping the quality and continuity of service for many users. Secondly, once the IP layer detects the fault, the recovery actions triggered may interfere with those at lower layers, which had detected the failure much earlier but did not have time to complete the restoration [10]. If the restoration is left to the IP layer, the detection and localisation of the failure may be much longer, due to the timer scales used (hello and keep alive timers) [12]. Generally, the timing aspect is an issue for layered networks. A reference to the timing problem can be found in [4]. The paper summarizes issues related to IP over optical networks, and gives some timing parameters to perform a coordinated end-to-end restoration.

End-to-end survivability generally relies upon a end-to-end management control of the layered network. The coordination of the recovery mechanisms often requires the intervention of an integrated management system. Some failure scenarios cannot be effectively resolved without the intervention of an integrated management [19]. Correlation between alarms at different network layers are

necessary. The project MISA refers a model for the ATM/SDH case. Standardisation of protocols and a generic approach are required for implementation in real networks.

Due to the undesirable detection of a physical failure at different layers, the TMN might not locate the origin of the fault. Upon a failure at the physical layer, the propagation of the failure might preclude the TMN from locating the failure between the physical layer and the cross-connect of higher layers [11]. The issue is to prevent the activation of the recovery mechanisms at higher layers when only the lower layer recovery is needed.

Another consideration is that the optical layer cannot, in general, detect faults at higher layers. Therefore, that layer might not be able to provide a true protection [27]. The integrated TMN, for instance, could inform the optical layer in order to trigger the recovery mechanism at this layer.

The coordination between restoration mechanisms at different layers is possible by means of an escalation of the restoration between layers. Escalation of restoration at different layers is sometimes based on the use of hold-off timers, which prevent from duplicating and overlapping recovery actions. The use of an hold-off timer may, in some cases, slow down the restoration process. The resulting delay, when a hold-off timer is used for the coordination between the recovery mechanisms at the different layers could be annulled whenever the server layer recovery has failed. A proposition of adding new OAM signals from the server to the client layer has been made in [1], to neutralise the hold-off time in order to immediately trigger the recovery at the client layer.

The use of an hold-off timer does not always prevent a fault to propagate in higher layers. When the optical layer detects a fault, it cannot prevent the fault from being propagated to higher layers, causing the activation of the recovery mechanisms at these layers when these are not required. Such a case has been demonstrated in SDH/WDM networks [11,30]. This situation should be avoided to prevent deadlock.

Also, albeit the use of hold-off timer could resolve the problem of contention, it causes the completion of the restoration to be delayed. An issue is to determine if the standardisation of hold-off timers, as it had been done for SDH rings networks, is useful for other technologies or not.

Another unsolved issue applies to the "highest layer recovery" scheme. This strategy, detailed in previous sections, enables a single restoration mechanism for the traffic generated at different layers. Therefore, the restoration of affected traffic generated at a higher layer and carried over trunks at a lower layer, relies upon the restoration scheme of the higher layer. The issues arises when the backup path at the higher layer must be found. Correlation of the routing tables between the server and client layers to have client working and backup paths physically disjoint is necessary. If dynamic routing is considered, protocols must enable the correlation.

Finally, the last type of unsolved issue lies in the use of spare resources to enable the end-to-end restoration.

In order to minimise the cost of resources, the concept of sharing capacity be-

tween layers was developed. Also, grouping the connections at the higher layer could minimise the number of alarm generations, and speed up the restoration [32,33]. How is this grouping traded-off against the spare capacity requirements, i.e the extra cost, resulting from a consideration of a coarser protection granularity?

In most approaches the spare capacity is calculated in a top-down approach. A feedback from lower layers should be considered when the higher layers requirements are computed. This would enable the optimisation of spare resources. A feedback loop was used in some cases during PANEL but its complexity and run-time performance makes it difficult to use. New mechanisms should make the feedback loop more attractive.

Moreover, an additional cost to the transmission spare capacity must be considered: the cost varies when different ports to separate spare and working resources are used [33]. The separation of working and spare capacity enables the sharing spare capacity between restoration strategies at different layers, at the expense of an additional port cost. An evaluation of the spare capacity savings compared with the cost of additional ports would be interesting.

4 Conclusion

Different strategies of providing end-to-end survivable multilayer networks have been discussed. From the results of the experiments, comparisons of the different strategies were possible but no single “best” strategy can be derived. Further studies are necessary to consider different degrees of integration of a technology on top of others.

The problem of survivability in multilayer networks has been well defined and the main issues have been identified. Some strategies have affected the work of the Standards organisations, producing the release of various guidelines.

Nevertheless, some issues are still unsolved, leaving the problem of survivability in multilayer networks open. Three types of issues have been highlighted: detection of the failure, management of restoration mechanisms, and efficient use of spare resources. The detection of a failure must be prompt to enable a fast restoration of the affected traffic. Since long time spells between messages, informing about the state of the links, occur in IP networks, it is not currently feasible to achieve a fast restoration only relying on the restoration at this layer. Moreover, management issues arise when re-routing strategies are used in the end-to-end survivability implementation. The issues consider the use of an integrated management system, that requires standards, as well as correlation mechanisms between topologies at different layers when the restoration is left to higher layers. Finally, multi-layer restoration implies a cost in resources, that is not also a transmission capacity cost but also a extra cost in the number of ports. Comparisons of restoration strategies that include the cost of the nodal cost (ports and backplanes) would be of interest.

References

1. Autenrieth, A., Kirstadter, A.: Fault Tolerance and Resilience Issues in IP-Based Networks. Second International Workshop on The Design of Reliable Communications Networks (DRCN). Munich. April 2000
2. Autenrieth, A., Van Doorselaere, K., and Demeester, P.: Evaluation of Multilayer Recovery Strategies. Eunice'98. Open European Summer School. Munich. Germany. Sept.1998
3. Baroni, S.: Backbone Network Architectures for IP Optical Networking. Phil. Trans. R. Soc. Proceedings. pp 2233-2254. London. 2000
4. Chandhok, N., Duresi, A., et al: IP over Optical Networks: A Summary of Issues. Internet Draft. July 2000
5. Demeester, P., et al.: Resilience in a Multilayer Network. Invited lecture at the 1st International Workshop on the Design of Reliable Communication Networks (DRCN). Brugge, May 1998
6. Demeester, P., et al.: Network Resilience Strategies in SDH/WDM Multilayer Networks. ECOC'98. Madrid. 20-24 September 1998
7. Demeester, P., and Gryseels, M., et al.: Resilience in Multilayer Networks. IEEE Communications Magazine. Vol.37. No.8. August 1999
8. Doverspike, R., Phillips, S., Westbrook, J.: Future Transport Network Architectures. IEEE Communications Magazine. Vol.37 No.8 pp.96-101. August 1999
9. Draft Technical Report: Enhanced Network Survivability Performance. T1A1.2 Working Group. November 2000
10. Ester, G., T'Joens, Y., Vandenhouste, M.: Restoration Schemes For Optical And SONET/SDH-Based IP Networks. Alcatel Telecommunications Review. 2nd Quarter 1999
11. Overall Network Protection - Final Report. ACTS-PANEL. Deliverable D3a. January 1998
12. Ghani, N., Dixit, S., Ti-Shiang, W.: On IP-over-WDM Integration. IEEE Communications Magazine. March 2000
13. Gryseels, M., Clemente, R., Demeester, P.: Protection Strategies for SDH-over-WDM networks. Proc. NOC'98. Manchester. UK. June 1998
14. Gryseels, M., Demeester, P., Clemente, R., and Ravera, M.: Common Pool Survivability for ATM on SDH Ring Networks. Proc. 8th Int'l. Symp. Network Planning. Sorrento. Italy. Oct.1998
15. Gryseels, M., Struyve, K., Pickavet, M., and Demeester, P.: Survivability Design in Multilayer Transport Networks. Proc. ICTSMA. Nashville. TN. USA. pp. 535-48. 5-8 March 1998
16. Harada, Y.: AC205 PANEL Report: Recommendation For Multiple Recovery Scheme. ACTS FORUM Reform Day in NTT. Tokyo. November 1998
17. ITU-T Rec. G.803: Architectures of Transport Networks Based on the Synchronous Digital Hierarchy (SDH). 1993
18. ITU-T Rec. G.841: Types and characteristics of SDH network protection architectures. December 1997
19. Jajszczyk, A., Brianza, C., Janukowicz, D.: TMN-based Management of Multi-layer Communications Networks. 1st International Workshop on the Design of Reliable Communication Networks (DRCN). Brugge. May 1998
20. Johnson, D.: Survivability Strategies for Broadband Networks. Proc. IEEE Globecom '96. London. pp.452-6

21. Jones, M. L. et al.: Sprint Long Distance Network Survivability: Today and Tomorrow. *IEEE Communications Magazine*. Vol.37 No.8 pp.58-62. August 1999
22. Kawamura, R.: Architectures for ATM Network Survivability and Their Field of Deployment. *IEEE Communications Magazine*. Vol.37 No.8 pp.88-94. August 1999
23. Manchester, J., Bonenfant, P., Newton, C.: The Evolution of Transport Network Survivability. *IEEE Communications Magazine*. Vol.37 No.8 pp.44-50. August 1999
24. Metz, C.: IP Protection and Restoration. *IEEE Internet Computing*. Vol.4. No.2. March/April 2000
25. Neoderlof, L., Struyve, K. et al.: End-to-End Survivable Broadband Networks. *IEEE Communications Magazine*. Vol.33 No.9 pp.63-70. September 1995
26. Noh, T.: End-to-End Self-Healing SDH/ATM Networks. *IEEE Globecom '96*. New York. USA. pp.1877-81 vol.3
27. Owens, K.: Network Survivability Considerations for Traffic Engineered IP Networks. ietf draft. draft-owens-te-network-survivability-00.txt. March 2000
28. Struyve, K., Demeester, P.: Escalation Between Recovery Schemes: Beyond ATM Backup Recovery. *Proc. Globecom'96*. pp.755-61
29. Struyve, K. et al.: Application, Design, and Evolution of WDM in GTS's Pan-European Transport Network. *IEEE Communications Magazine*. Vol.38 No.3 pp.114-121. March 2000
30. Van Doorselaere, K. et al.: Fault Propagation in WDM-based SDH/ATM Networks. 1st International Workshop on the Design of Reliable Communication Networks (DRCN). Brugge. May 1998
31. Veitch, P. et al.: An Integrated Restoration System for SDH-based ATM Transport Networks. *IEEE Globecom'96*. New York. USA. pp.1882-6 vol.3
32. Veitch, P., Johnson, D.: ATM Network Resilience. *IEEE-Network*. Vol.11. No.5. pp.26-33. Sept-Oct 1997
33. Veitch, P., Johnson, D., Hawker, I.: Design of Resilient Core ATM Networks. *Proc. IEEE Globecom '97*. New York. pp.1450-4
34. Wu, T.: Emerging Technologies for Fiber Network Survivability. *IEEE Communications Magazine* Vol.33 No.2 pp.58-74. 1995
35. Ye, Y., Dixit, S., Ali, M.: On Joint Protection/Restoration in IP-centric DWDM-Based Optical Transport Networks. *IEEE Communications Magazine*. Vol.38 No.6 pp.174-183. June 2000

Author Index

- Abolhasan, M. II-186
Ahn, G. II-319
Akimaru, H. I-65
Al-Ibrahim, M. II-239
Al-Regib, G. II-745
Ali, M.M. I-429
Altman, E. II-97
Altunbasak, Y. II-745, II-786
Amorim, C. II-776
Anagnostopoulos, C. I-459
Anagnostopoulos, J. I-459
Anisimov, N. II-559
Ano, S. II-825
Antonakopoulos, T. II-497
Antonov, V. II-559
Araki, T. II-620
Asfour, T. I-589
Ashraf, S.N. II-805
Atmaca, T. I-107, I-348
- Balbinot, L.F. I-204
Banerjee, G. II-302
Barakat, C. II-97
Bartoš, R. I-388
Bartz Ceccon, M. II-277
Becker Westphall, C. II-438
Beghdad, R. I-520
Benaboud, H. I-780
Bester, J. II-348
Bettahar, H. I-630
Blanpain, Y. II-259
Boavida, F. II-458
Bonnet, C. II-31
Bordes, J. I-124
Bosquiroli Almeida, M.J. II-277
Bouabdallah, A. I-630
Bouzida, Y. I-520
Boyle, J. II-559
Braun, T. II-206
Brogle, M. II-206
Budkowski, S. I-649
- Cáceres Alvarez, L.M. II-438
Caminero, M.B. II-358
- Camps, J. II-648
Capelle, A. Van de I-790
Carrapatoso, E. I-27
Carrión, C. II-358
Čavrak, I. II-399
Chang, S.W. I-709
Chao, H.-C. I-307, I-709
Chatterjee, M. I-157
Che, H. I-368
Chen, H.-C. II-225
Chen, J.-L. I-307, I-709
Chen, Y.C. II-82
Cheng, H.-C. I-307
Cherkasova, L. II-71
Choel, B.-Y. I-184
Choi, H. I-698
Choi, I. I-698
Choi, M.C. II-137
Choi, S. II-61
Choi, Y. I-771
Choi, Y.-I. II-380
Choplin, S. II-527
Chow, C.E. I-753
Chow, C.O. II-687
Chujo, T. I-753
Chun, W. II-319
Chung, J.-M. II-735
Čičić, T. I-488
Citro, R. II-538
Colcher, S. II-867
Corral, G. II-648
Czachórski, T. I-107
- Das, S.K. I-157
Dastangoo, S. I-336
Davis, C. I-124
Davis IV, N.J. II-370
Decre, N. II-825
Demestichas, P. I-286
Deniz, D.Z. II-570
Denz, P.R. I-449
Dey, D. I-326
Di, Z. I-245
Dimiyati, K. II-687

- Droz, P. I-218
 Duato, J. II-358
 Dutkiewicz, E. I-720, II-186

 Ehrensberger, J. I-378
 Ehsan, N. II-51
 Eissa, H.S. II-268
 El-Sayed, A. I-610
 Elbiaze, H. I-348
 Estepa, R. II-549

 Fabbri, A. I-620
 Fan, L. I-147
 Fan, Z. I-55
 Fazekas, P. I-296
 Fernández, D. II-697
 Fornés, J.M. II-449
 Freire, M.M. I-358
 Fujii, A. II-845
 Fulp, E.W. I-409

 Gardiner, J.G. I-147
 Garrell, J.M. II-648
 Gaspary, L.P. I-204
 Ghosh, S. II-538
 Gjessing, S. I-488
 Glitho, R. II-707
 Go, S. II-639
 Gomes, A.T.A. II-867
 Goossens, I.G. II-795
 Goossens, I.M. II-795
 Graja, H. I-803
 Günter, M. II-206
 Guerra, S. II-697
 Gupta, A.K. II-815
 Gutacker, D. II-590

 Ha, E.-J. II-726
 Hämäläinen, T. II-250
 Hagino, T. II-177
 Hamadi, R. II-707
 Han, D.H. II-517
 Haque, A.U. II-628
 Harle, D. I-829
 Hasegawa, T. II-825
 Hashmani, M. I-398
 Hava Muntean, C. I-821
 Hayashi, M. II-31

 Hayes, J.F. I-429
 He, J. I-753
 He, X. I-368
 Heemstra de Groot, S. I-326
 Hegde, M.V. I-124
 Hei, X. I-45
 Hendler, M. I-478
 Hladká, E. II-612
 Hoang, D.-H. I-599
 Hoang, T.T.M. I-743
 Homan, P. II-348
 Hommel, G. I-478
 Hong, W.-K. II-599
 Hsieh, H.-Y. I-569, II-259
 Huang, C.-M. I-559
 Huie, R. II-707

 Ikenaga, T. I-398
 Ilyas, M. II-677
 Imre, S. I-296, II-11, II-468
 Iqbal, T. II-570
 Ishikawa, E. II-776

 Jäger, R. I-419
 Jakobs, K. I-194
 Jedruś, S. I-107
 Jones, G.A. II-766
 Jun, K.-P. II-380
 Jung, D.-S. I-1
 Jung, M.-J. II-599

 Kaario, K. II-250
 Kabatnik, M. II-20
 Kambo, N.S. II-570
 Kamel, T. II-268
 Kandel, O. I-657
 Kanter, T. I-12
 Kesidis, G. II-339
 Khan, J.I. II-628
 Khotimsky, D.A. I-137
 Kim, C. II-61
 Kim, K. I-117
 Kim, M. II-786
 Kim, S.-C. II-735
 Kim, Y.-C. II-1
 Kinicki, R. I-98
 Klöcking, J.-U. I-498
 Ko, J.-W. II-1

- Kobayashi, T. I-167
 Konstantinopoulou, C. I-286
 Koutsopoulos, K. I-286
 Kovacevic, V. I-75
 Kovačić, D. II-399
 Kuhlins, S. II-590
 Kung, H.-Y. I-559
 Kuo, S.-Y. I-317
 Kure, Ø. I-488
 Kwon, J.-H. II-409, II-726

 Laalaoua, R. I-107
 Lac, C. II-805
 Lau, C.-T. II-168
 Lavery, B. I-37
 Leal, J. II-449, II-549
 Lee, B.-J. II-1
 Lee, B.-S. II-168, II-380, II-815
 Lee, C.L. II-82
 Lee, D.-E. II-1
 Lee, S.-H. I-1
 Lee, Y. I-771
 Leiss, E.L. I-549
 Li, X. I-235
 Liao, H.Y. II-82
 Liao, W.-H. II-158
 Libeskind-Hadas, R. I-508
 Lin, F.Y.-S. II-148
 Linwong, P. II-845
 Liu, M. II-51
 Lorenz, P. II-422

 Magoni, D. I-762
 Maher, M. I-124
 Maihöfer, C. I-498
 Makios, V. II-497
 Mammeri, Z. II-422
 Mandyam, G.D. I-157
 Maniatis, S. II-108
 Manjanatha, S. I-388
 Marbukh, V. II-309
 Marias, G.F. I-439
 Marques, P. II-458
 Marroun, E. II-735
 Masuyama, H. II-877
 Matsikoudis, E. I-286
 McClellan, S. II-579
 McManis, J. I-803, I-821

 Melhem, R. I-508
 Merabti, M. II-507
 Merakos, L. I-439
 Michalareas, T. I-687
 Mikou, N. I-780
 Milanovic, S. II-717
 Miller, L.K. I-549
 Miloslavski, A. II-559
 Mirchandani, V. I-720
 Miyoshi, T. II-877
 Mokhtar, H.M. II-507
 Mouchtaris, P. I-117
 Mouftah, H.T. I-245
 Mouharam, A.A.K. II-658
 Müller, F. I-276
 Muntean, G.-M. I-540
 Murphy, J. I-821
 Murphy, L. I-540
 Mutka, M.W. I-640, II-71

 Nakamura, T. II-620
 Naraghi-Poor, M. I-124
 Nemoto, Y. II-845
 Nguyen, H.N. I-37
 Nguyen, H.-T. I-256
 Nikolouzou, E.G. I-266
 Nilsson, A.A. I-449

 Ogino, N. I-673
 Oie, Y. I-398
 Oktug, S.F. II-390
 Oliveira, J. I-27
 Onozato, Y. I-579
 Owen, H.L. II-137

 Pagani, E. I-468
 Palola, M. II-478
 Pansiot, J.-J. I-762
 Papadakis, A.E. II-488
 Papadopoulos, A. II-497
 Parish, D.J. II-766
 Park, C.G. II-517
 Park, H.-J. II-380
 Park, J.S. II-370
 Park, J.-T. II-409, II-726
 Park, S.-W. I-1
 Perdikeas, M.K. II-488
 Pereira, R. II-507

- Petrovic, Z. II-717
 Pham, H. I-37
 Phan, S. II-836
 Pieprzyk, J. II-239
 Pils, C. I-194
 Pinto Lemos, R. II-668
 Pletka, R. I-218
 Pogosyants, G. II-559
 Pollak, A. II-890
 Popovic, M. I-75
 Potemans, J. I-790
 Protonotarios, E.N. I-459
 Protopsaltis, N.G. I-459
 Pyrovolakis, O.I. II-488

 Quiles, F.J. II-358

 Radhakrishnan, S. I-184
 Räisänen, V. II-127
 Rao, N.S.V. I-184
 Rapp, J. I-727
 Rashid, E. II-620
 Reeves, D.S. I-409
 Renno, J.R. II-766
 Reschke, D. I-599
 Rhee, S.H. I-698
 Rindborg, T. I-12
 Risso, F. I-85
 Roca, V. I-610
 Rockenbach Tarouco, L.M. I-204,
 II-277
 Rodriguez, J. II-422
 Roque, R. I-27
 Rossi, G.P. I-468
 Rothermel, K. I-498
 Roussev, B. II-854
 Russell, L. II-71
 Rzehak, H. I-256

 Sacks, L. I-687
 Sahlin, D. I-12
 Saleh, H. I-811
 Salvador, M.R. I-326
 Salvat, Z. II-612
 Sampatakis, P.D. I-266
 Samtani, S. I-117
 Sandhu, H. II-735
 Saydam, T. II-329
 Scharinger, J. II-196

 Schmid, A. I-657
 Schmid, O. I-124
 Sedillot, S. I-27
 Seet, B.-C. II-168
 Sekkaki, A. II-438
 Serhrouchni, A. I-589
 Shaikh, F.A. II-579
 Shamsuzzaman, M. II-815
 Shao, H.-R. I-235
 Sheu, J.-P. II-158
 Shinagawa, N. I-167
 Shiraishi, Y. I-579
 Shkrabov, S. II-559
 Sidhu, D. II-302
 Silva, H.J.A. da I-358
 Silva, J. II-458
 Silva, L. II-458
 Simões, P. II-458
 Singh, A.K. II-217
 Sivakumar, R. I-569, II-259
 Smith, H. I-640
 Soares, L.F.G. II-867
 Sokol, J. II-137
 Solís Barreto, P. II-668
 Steigner, C. I-657
 Stiller, B. I-218
 Storch, R. I-204
 Stranjak, A. II-399
 Stuckmann, P. I-276
 Sue, C.-C. I-317
 Sugar, R. II-468
 Sung, K.-Y. I-530
 Suzuki, M. I-673
 Szabo, S. II-11

 Takahashi, Y. I-167
 Talpade, R. I-117
 Tanaka, Y. I-65
 Tang, W. II-71
 Taniguchi, H. II-877
 Tassioulas, L. II-339
 Tatsuya Watanabe, W. II-438
 Templemore-Finlayson, J. I-649
 Terner, J.A. II-549
 Teughels, M. I-790
 Theologou, M.E. I-286, I-459
 Theunis, J. I-790
 Tizraoui, A. I-811
 Touvet, F. I-829

- Tsang, D.H.K. I-45
 Tsaur, W.-J. I-174
 Tseng, Y.-C. II-158
 Tsetsekas, C. II-108
 Tsykin, M. II-291
 Tunncliffe, M.J. II-658, II-766
 Tutsch, D. I-478

 Umemoto, M. I-737
 Urien, P. I-811

 Van Lil, E. I-790
 Venieris, I.S. I-266, II-108, II-488
 Vergados, D.D. I-459
 Vinyes, J. II-697
 Vozmediano, J.M. II-549

 Wallbaum, M. I-194
 Wang, S.-L. II-158
 Wehrle, K. II-117
 Wendt, F. I-204
 Wolfinger, B.E. II-754
 Wong, J.W. II-639
 Wong, L. I-117

 Woo, M. II-41
 Woodward, M.E. I-147
 Wu, H. I-235
 Wu, J. II-854
 Wu, Z. II-639

 Wu, L.-C. II-225
 Wysocki, T. II-186

 Yalamanchili, S. II-358
 Yamamoto, U. I-579
 Yamori, K. I-65
 Yang, C. II-836
 Yang, J. I-753
 Yang, L. I-640
 Yasuda, K. II-177
 Yegoshin, L. II-559
 Yen, H.-H. II-148
 Yoon, J. I-698
 Yoshida, M. I-398
 Yuan, X. II-677
 Yucel, S. II-329

 Zaballos, A. II-648
 Zaddach, M. II-754
 Žagar, M. II-399
 Zambenedetti Granville, L. II-277
 Zappala, D. I-620
 Zhang, J. II-250
 Zhang, X. I-429
 Zheng, Z. I-98
 Zhou, J. I-579
 Zhuklinets, I.A. I-137
 Zorn, W. I-743
 Zugenmaier, A. II-20